

Published Bi-Monthly

Vol. 1 #4 Mar./Apr. 1988

\$2.50 U.S.A.

\$3.50 Canada

CoCo Clipboard Magazine

THE NEWEST, MOST INDEPTH MAGAZINE FOR TANDY'S COLOR COMPUTER 2 & 3

Reviews
on

RGB
Hard Disk System
COMPATIBLE WITH RS-DOS

1987
CoCo Federal Tax

VIP
Writer
III

And
We Announce...

ClipDisk



FROM THE DESK OF...

Ted & Darlene Paul

That's it. I've had it! I've started this column at least a million times and the last one Darlene said was a "yawn and a half!" It's now too late at night, too cold outside and I've got more stuff to yap about than a politician in Des Moines tomorrow morning. So here goes, the last re-write before the printer, no interruptions please.

News from Tandy! The Multi-pak Interface is S.O.W.G. just like the RS232C Pak. A statement from Tandy says that there are no "immediate" plans to replace the unit, and that there is an 18 month supply in the retail chain. Yep, and there wasn't going to be any CoCo 3 either, and no laser printer. Okay, I can't really beef about this too much. The folks at Tandy have been pretty good to us here at Clipboard and I don't want to return evil for good, but what do you do with a statement that sounds like it came from the C.I.A.? Guess we'll wait. In the meantime if you need an MPI grab one at discount, get the CoCo 3 fix and have some fun. Personally I can wait till those "immediate" plans turn up on the shelf.

New Products! New Products! and more new products. Sure, I know, I've heard the rumors before, no more Epyx software, big bailouts at other magazines, no MPI, CoCo owners calling landfills to find out about mass disposal. HOG WASH! P.T. Barnum made a lot of money charging \$.50 to pass through a door marked "This way to the Egress." Take the rumors through that door will ya pal, the CoCo isn't dead, it isn't sick, it's not a slow mover out of Tandy's warehouses or factories either. The CoCo market is about to kick some - oops - make that ruffle some, ah let's say make some big waves - er ah. Trying to be diplomatic at this juncture is not easy. The CoCo is the hottest 8 bit machine on the market today for two reasons - it's got a hot chip 68B09E and two solid operating systems, RSDOS and OS9 Level II. Let's not get all bent out of shape here, I know RSDOS isn't really an "OS" but some people don't know that and most people don't care. If it works, buy it, use it, make it productive. And that's just what some very rational people have done.

Our front cover shows 4 new items for the CoCo. First is Tandy's Home Publisher program for the CoCo 3. It's no Aldus Page Maker program for several reasons, no rules, no leading and no real type fonts. But it doesn't cost like Page Maker does either and it is slick, runs pretty darn fast and it's Multiview

compatible! For company newsletters and club newsletters it is great and it supports more than just Tandy printers to boot. It's the best Tandy produced (written by Spectral Associates for Tandy) program for the CoCo 3 yet. It's a sure buy for many and if someone comes up with fonts that look like real typesetting you let me know and I'll publish the code here in Clipboard.

Second and third items; VIP Writer III and RGB Hard Disk drives. VIP III is dynamite for several reasons. First it is available right now. It has 40 and 80 column screens, a 49k print spooler and VIP Speller all for one reasonable price. Look out for SD Enterprises, they've got the whole VIP series under license for the CoCo and they're raring to go. On top of that RGB Computer Systems has got a hard disk system for your CoCo that is 100% RSDOS compatible. I saw it run CoCo Max, VIP Writer III, GETerm and a host of other CoCo programs. RGB can quote you a price for any sized HD you'd like and the thing works like lightning. A 40 meg. drive is equivalent of 255 floppies on line and ready to go. A 20 meg. system gets you about half that amount. RGB can partition the drive into RSDOS and OS9 any way you want. RGB is also talking with SD Enterprises and you just might see the VIP line be compatible with the RGB HD system. Wow! Imagine doing a disk directory from the command line of VIP III and asking for drive 235, or 13 or 128 and seeing the contents pop up on the screen in microseconds. Ya, sure the CoCo is dead - baloney!

Last but not least is the bargain of the year program from Puritas Springs Software, CoCo Federal Tax. This huge package comes with three disks, 100+ pages of instructions and will walk you through your 1040 and schedules A, B, C, D etc., etc. in no time flat. It runs fast and is just about idiot proof. This is the fifth year for Puritas Springs to bring out a CoCo Federal Tax program, and considering all the changes that have been made this year is truly a new program for the CoCo.

We've been co-sponsoring a monthly conference on CompuServe on the second Saturday of each month since January. Yes, Billy I know there is another database service and they

Continued on page 21

Editor and Publisher
Theodore W. Paul, Jr.

Associate Editor
Darlene M. Paul

Contributing Writers:

Rush T. Caley, Jim DeStefano,
Kraig Brockschmidt, Paul
Bornemann, Mike Dooley, Bob van
der Poel, Delmar Searles, Randy
Krippner

CoCo Clipboard Magazine™
is published bi-monthly (6 times
a year) by:
CoCo Clipboard Magazine
3742 U.S. 20, Box 3
Fredonia, N.Y. 14063
(716) 679-0126

ENTIRE CONTENTS COPYRIGHTED

**NO PORTION OF THIS MAGAZINE MAY
BE REPRODUCED, BY ANY MEANS,
EXCEPT BY EXPRESS WRITTEN
PERMISSION OF THE PUBLISHER.**

Tandy, Extended Color Basic,
Color Basic and Program Pak are
registered trademarks of Tandy
Corp. CompuServe is the
registered trademark of
CompuServe, Inc.

All programs, information and
articles are distributed on an
"as is" basis without any
warranty of any kind whatsoever.

CoCo Clipboard, The Clipboard,
Clipboard BBS Systems, CCMDisk
Service, "Clipper Koala" and
Clipboard Convo are trademarks
of CoCo Clipboard Magazine.

Subscriptions to CoCo Clipboard
magazine are \$12.00 per year for
6 issues, \$23.00 for 2 years -
12 issues. We accept checks or
money orders only. Canadians may
pay \$12.00 in U.S. funds.
\$18.00 /year after 8/1/88. All
overseas subscriptions are
\$25.00 surface mail. Prices good
till 8/1/88

Orders may be sent to:
CoCo Clipboard Magazine
3742 U.S. 20 Box 3
Fredonia, N.Y. 14063

Subscription orders received
after the first day of a
publication month will begin
with the next issue. We do not
start subscriptions by mailing
back issues.

Full refund after one issue. No
refunds after two issues.

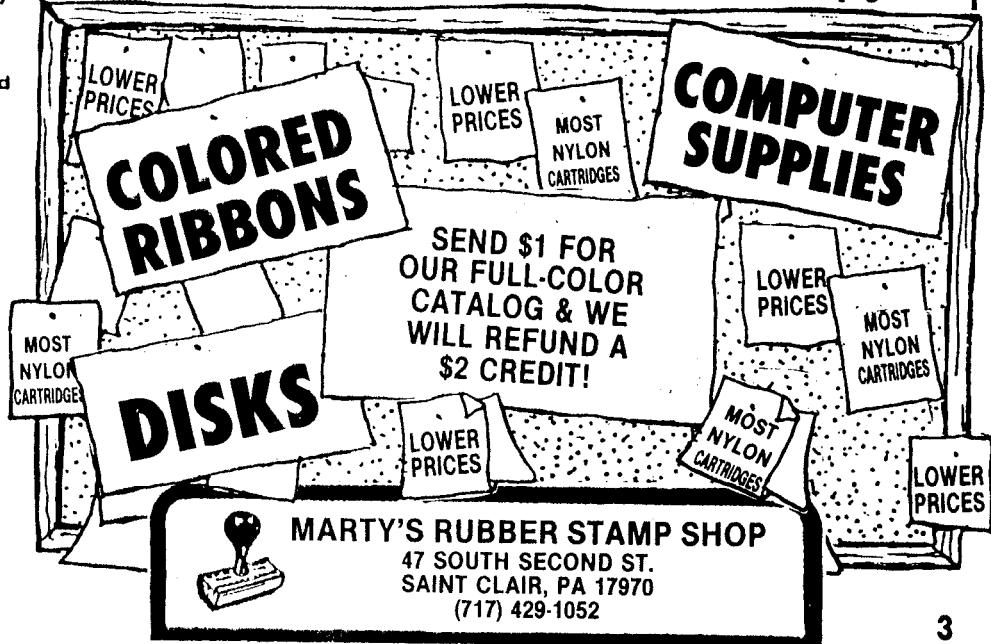
Your comments and suggestions
are welcome. We reserve the
right to edit and publish all
letters received unless
requested not to by the writer.

CoCo Clipboard Magazine

Vol. 1 Issue 4 March/April 1988

TABLE OF CONTENTS

From the Desk of by Ted and Darlene Paul	inside front cover
Pascal Programming by Delmar Searles	page 4
The 9th. Power by Randy Krippner	page 9
Basic Help by Bill Bernico	page 13
A Better Display by Jim DeStafeno	page 14
DataBase Tutorial Pt. II by Rush Caley	page 17
Product Reviews by Paul Bornemann, Lee Parker, Ted Paul	page 23
The Assembly Line by Kraig Brockschmidt	page 33
CoCo 'N Amateur Radio by Mike Dooley	page 38
BLIST for BASIC09 by Bob van de Poel	page 41
* * * * *	
CompuServe Conference Information	page 20
Subscription Information	page 35
Back Issue Information	page 39
Advertising Information	page 20
CoCo Club Registration Form	page 44
Programming Corrections	page 45
Clipboard Coupons	page 36



Pascal Programming

Delmar Searles

Two close friends want to divide equally the contents of an eight-gallon water jug. The jar is full and the only other containers available are two empty water jugs. One holds five gallons and the second holds three gallons. None of the jars have any markings inside or out. How can the friends divide the water evenly?

One possible approach is the "trial and error" approach. Start pouring one jar into another until you arrive at the desired goal. Before attempting this we must devise a method of keeping track of how much is in each jug and we must determine exactly what our goal is.

If you think about it, our goal must be to get exactly four gallons of water into each of the first two jugs. Remember that the third jug only holds three gallons. As to keeping track of each jug's contents we can proceed as follows (for example):

	Jug 1	Jug 2	Jug 3
Start	8	0	0
Pour jug 1 into jug 2 (Jug 2 only holds 5 gallons)	3	5	0
Pour jug 2 into jug 3	3	2	3
Pour jug 2 into jug 1	5	0	3
Pour jug 3 into jug 2	5	3	0
Pour jug 1 into jug 3 and so on....	2	3	3

Chances are we would eventually find a solution if we kept this process up long enough.

If we are interested in finding the shortest possible solution or in listing all of the solutions, the trial and error approach is inadequate. Instead we would have to devise some systematic plan of attack. That is, we would have to create an algorithm for finding all solutions (or at the very least for finding the shortest solution).

Let me use the word "state" to refer to the contents of the three jugs at any particular point in time. A "step" or "action" is

the pouring of one jug into another. Given any state I will perform these tasks:

```
if water is equally divided then
  print list of steps showing the
  solution to puzzle
else if current state is NOT a repeat then
```

```
  try pouring jug 1 into jug 2
  try pouring jug 1 into jug 3
  try pouring jug 2 into jug 1
  try pouring jug 2 into jug 3
  try pouring jug 3 into jug 1
  try pouring jug 3 into jug 2
endif
go back to previous state
```

From each state I will always try all six actions leading to another state. In some cases I may not be able to perform the indicated action. For example, if jug 3 is already full then I cannot pour either jug 1 or jug 2 into jug 3.

After performing a given action I check to see if I'm done. If so, I record the steps that brought me to this solution. Then I back up to the previous state and try another action.

After any given step I also check to see if I've encountered the same state before in the steps I've taken to get there. If I have, then I'm just going in circles. I immediately back up to the previous state and try the next possible action.

If I've not solved the puzzle and I'm not going in circles then I methodically try each of the six steps to see if they lead to a solution from the current state. If you try this on paper, you'll see that you have to do an awful lot of backtracking and trying to remember which actions you've tried at various states in the current path of steps taken.

A problem such as this can be solved quite easily by using a recursive procedure (a procedure that calls itself). When a procedure terminates, program control returns to the statement immediately following the procedure call. This is true even if a procedure calls itself. In the algorithm described above, the phrase "go back to the previous state" means terminate the procedure and try the next action in the previous state.

In general, a recursive procedure must have two properties. The first is a termination condition. In our problem there are two such conditions: we've reached a solution or we're going in circles. The second property is that the procedure must do something that leads to the termination condition. In solving the puzzle, we always pour the contents of one jug into another. This action is one more step towards solving the problem, or towards discovering that we are going in circles.

The procedure NextStep in the program WatJugs is based on the algorithm described above. There are a few preliminary bookkeeping details that must be handled before performing the algorithm. The procedure is called by identifying which jug (A) is to be poured into which other jug (B) where A and B are 1, 2, or 3. The procedure first performs the "pouring". It pours as much as possible of jug A into jug B.

When the pouring is complete, the program records the action or step that was taken and the current state that resulted. Each is added to a corresponding list that records all of the steps and the resulting states along the current path.

If the current state is the desired state (four gallons in each of jug 1 and jug 2), then the procedure calls a PrintPath procedure to print out the steps and corresponding states that lead to the solution. In that case, the procedure terminates and goes back to the previous state. Notice that whenever the procedure terminates, it removes the current step and state by decrementing the NumSteps (number of steps) counter. This returns the lists to their previous condition

☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺
 ☺ Dear Reader:
 ☺ We really don't have room in this small ad to list all our great products and even greater prices--utilities for RS-DOS and OS9, mailing list programs, a great crib game, editors, Ultra Teletype, Lister, Packer, Ultra Label Maker, Magazine Index System, etc.--so why not do your CoCo a favor and get our latest newsletter. Normally we like to get a buck for postage, but for our CoCo Clipboard friends we're going to make an exception. Just send us your name and address and we'll send you our informative newsletter loaded with hints, tips and products.
 ☺
 ☺ Bob van der Poel Software
 ☺ 17435 - 57 Avenue
 ☺ Edmonton, Alberta
 ☺ Canada T6M 1E1
 ☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺☺

(corresponding to the previous state).

If a solution has not been found the NextState procedure calls a boolean function to determine if the current state is a repeat of an earlier one. The function NewState returns TRUE if the current state is a new state and FALSE if not. It does this by searching the current list of states. If it finds a match with the current state the function returns FALSE, otherwise it returns TRUE.

There may be times when an attempted action (or step) is not possible. For example, if the second jug (jar B) is already full then no water is poured. In that case the new state is exactly the same as the previous state and we are going in circles. The program then immediately returns to the previous state and tries the next pouring step in the list of six.

Eventually the program will complete trying all six steps on the starting condition and the program will terminate. As it turns out, there are sixteen different solutions to the water jugs problem. The longest requires 15 steps (two different solutions). The shortest requires only seven steps:

R.A.D.
TEXTFORM

R.A.D. Products Presents TEXTFORM

Finally, a versatile text formatter is available for the Color Computer. TEXTFORM is compatible with all models with at least 64K, even the Color Computer III. This machine language program will format ASCII text files into two column pages quickly and easily. Text may be left unmodified, or simply insert special formatter commands for added control. TEXTFORM is a versatile enhancement to any word processing system whether you are a casual or professional user.

Software supports:

- Output to printer or disk
- Most popular printers
- Adjustable format parameters
- Columnar data
- Multiple page titles
- Optional page numbering
- Large files (up to a full disk)

TEXTFORM comes with complete documentation as well as sample format examples. Onscreen parameter display takes the guesswork out of format settings. Customized parameters may be saved to disk and reloaded for future use, thus eliminating mistakes and configuration time. Special printer codes and baud rate settings are software selectable. TEXTFORM is programmed in a high resolution environment which incorporates pull-down menus for ease of use. The software also supports auxiliary peripheral input from joysticks, mouse, touchpad, and high resolution input pack for added program control.

This is not another word processor. There are many fine word processors on the market for the Color Computer. TEXTFORM is a user definable two column text formatter. If you are looking for a program which will let your Color Computer be even more professional looking documents without hours of tedious work, then TEXTFORM is the answer.

Ideal for:

- School newspapers
- Club newsletters
- Business reports
- Bulletins
- Advertisements
- Program listings
- And much more...

System requirements:

- Color Computer (64K minimum)
- Disk drive
- Printer

TEXTFORM.....\$34.95

R.A.D. Products
198 Hotchkiss St.
Janestown, NY 14701
(716) 665-2124

Terms: Check, Money Order, C.O.D.
NY residents add 7% sales tax
C.O.D. orders add \$3.00
All orders add \$2.00 for shipping
All orders shipped within 24-48 hours
Express shipping available by request

Start:	800
Pour jug 1 into jug 2	350
Pour jug 2 into jug 3	323
Pour jug 3 into jug 1	620
Pour jug 2 into jug 3	602
Pour jug 1 into jug 2	152
Pour jug 2 into jug 3	143
Pour jug 3 into jug 1	440

The three digit number in each line indicates the current state after the indicated action. The first digit indicates the contents of the first jug, the second digit is the contents of the second jug, and so on. In the program, the steps are stored in a similar way except only two-digit numbers are used. The first digit indicates jug A and the second jug B. For example, step 23 means pour jug 2 into jug 3.

In the main driver of the program, NextState was called with the A and B parameters both set to one. This means pour jug 1 into jug 1. Obviously this will not be useful, but it does give the correct starting conditions on the initial call to NextState. When NextState has tried all six possible steps from the initial state, it will return control to the main driver and the program will terminate.

```

program MatrJugs(input, output);

type
  SmallArray = array [1..3] of integer;
  BigArray = array [1..24] of integer;

var
  Path, (* A history of past steps taken *)
  State: (* A history of the results of past steps *)
  BigArray;

  Capacity, (* Capacity of Jugs *)
  Contents: (* Amount currently in Jugs *)
  SmallArray;

  NumSteps: (* Number of steps taken *)
  integer;

  Printer: TEXT; (* Printer output *)

procedure PrintPath(N: integer);
(*
 * Prints the sequence of steps taken to reach
 * the solution of the puzzle.
 *)
var
  I: integer; (* A loop counter *)

begin
  writeln(Printer, 'Start:'':21, State[1]);
  for I := 2 to N do begin
    write(Printer, 'Pour jug ', Path[I] div 10 :1);
    write(Printer, ' into jug ', Path[I] mod 10 :1);
    writeln(Printer, State[I])
  end; (*for*)
  writeln(Printer);
  writeln(Printer)
end; (* PrintPath *)

function NewState(TestState: integer): boolean;
(*
 * Tests the next state after a given action to see
 * if its a new state or if we are repeating.
 *)
var
  I : integer; (* A loop counter *)

```

1987 COCO FEDERAL TAX

TAX PREPARATION SOFTWARE

64K CoCo 1,2or3 w/1 disk drive

Form 1040	Worksheets:
Schedule A	State/Local Tax
Schedule B	Pension/Annuity
Schedule C	Social Security
Schedule D	IRA
Schedule E	Dependents
Schedule SE	Earned Income Cr
Form 2441	Excess SocSoc
Form 4562	
Form 8615	

Menu driven, loads & saves to disk, prints to screen or forms acceptable to IRS, easy to use format follows IRS forms, comes with 100+ page owner's manual, built in calculator, checks for common errors and omissions, complete directory system for easy editing, disk directory function, release date 1/31/88

\$49.95

PURITAS SPRINGS SOFTWARE

Ameritrust Building
17140 Lorain Avenue
Cleveland, Ohio 44111
(216) 251-8085

Information on FREE 1986 Demo
upon request with S.A.S.E.

More 'GOODIES' from Bill Bernico Software

COGOPACK, the original 63 program disk with 21 fonts, music, graphics, utilities and more.

FUNPACK, which has expanded and additional font styles as well as COCOSIZE the exercise program for the Color Computer. (See April '87 Rainbow page 143 for the CoCoSize review).

VALUPACK, including dozens of additional, longer programs that wouldn't fit on COGOPACK.

3 PACK, a diskful of goodies exclusively written for the CoCo 3. More games, graphics, and useful, informative programs and programming hints and tips for your machine.

SUBPACK - Attention programmers. SUBPACK is a diskfull of handy and useful subroutines that will help make your programming projects a lot easier and shorten your programming time. Add any of these subroutines to your own programs.

Each disk is still only \$6.00. Cash, check or Money order only to: Bill Bernico Software 708 Michigan Ave. Sheboygan, WI 53081

We have what your Coco Max is starving for Art Deli

Art Deli is a set of 440 PMODE4 PICTURES ON 10 FLIPPY DISKS. EACH DISK CONTAINS HOLIDAY AND SEASONAL PICTURES THAT CAN BE LOADED INTO COCO MAX OR YOUR FAVORITE GRAPHICS EDITOR. GREAT FOR AIDING IN THE CREATION OF NEWSLETTERS, BROCHURES, ADS, OR DEMONSTRATIONS.

ART DELI CAN BE PURCHASED BY THE DISK OR FULL SET. THE FOLLOWING DISKS ARE AVAILABLE :

1) HALLOWEEN/THANKSGIVING
 2) CHRISTMAS 122
 3) TRAVEL/CRITTERS
 4) SPORTS/FUNNY SPORTS
 5) SUMMER/SPRING
 6) PEOPLE 122
 7) FOOD/MISC.
 8) WINTER/NEW YEARS
 9) EASTER/VALENTINE
 10) PATRIOTIC/CREATIVE ART

FOR COCO 1.2, & 3

\$12.95
 DISK
 ←
\$99.95
 SET*

*SET COMES
 W/PIX BOOK

"A SMORGASBORD
 OF IMAGES"
 RAINBOW 10.67

COMING SOON
 ART DELI II

**Specialty
 Projects**

4810 McCrory Ave.
 Memphis, TN 38122
 901-682-8737

ADD \$3.00 FOR SHIPPING AND HANDLING.
 CASH, CHECK, OR MONEY ORDERS ONLY !
 PLEASE, NO CODS OR CREDIT CARDS.

```

begin
  NewState := TRUE;
  for I := 1 to NumSteps - 1 do

    if TestState = State[I] then
      NewState := FALSE
  end; (* NewState *)

procedure NextStep(A,B: integer; Amount: SmallArray);
(*
 * This recursive procedure is the heart of the
 * program. See article for details.
 *)
var
  TransferAmount: integer;
  (* The amount transferred from jug A to jug B. *)
begin
  (* Find out how much we can pour from A into B. *)
  TransferAmount := Capacity[B] - Amount[B];
  if TransferAmount > Amount[A] then
    TransferAmount := Amount[A];
  (* Pour as much of jug A into jug B as possible. *)
  Amount[A] := Amount[A] - TransferAmount;
  Amount[B] := Amount[B] + TransferAmount;
  (* Update the history of steps taken and results. *)
  NumSteps := NumSteps + 1;
  Path[NumSteps] := 10*A + B;
  State[NumSteps] := 100*Amount[1]+10*Amount[2]+Amount[3];
  (* If water is evenly divided, print steps that did it. *)
  if (Amount[1] = 4) AND (Amount[2] = 4) then
    PrintPath(NumSteps);

  (* Otherwise, if you're not repeating states,
   go on to next steps *)
  else if NewState(State[NumSteps]) then begin
    NextStep(1,2,Amount); (* Pour jug 1 into jug 2 *)
    NextStep(1,3,Amount); (* Pour jug 1 into jug 3 *)
    NextStep(2,1,Amount); (* Pour jug 2 into jug 1 *)
    NextStep(2,3,Amount); (* Pour jug 2 into jug 3 *)
    NextStep(3,1,Amount); (* Pour jug 3 into jug 1 *)
    NextStep(3,2,Amount); (* Pour jug 3 into jug 2 *)
  end; (* else *)
  (* Exit and return to previous state. *)
  NumSteps := NumSteps - 1
end;

(*****
 * M A I N   D R I V E R
 ****)
begin
  (* Set up initial state *)

  NumSteps := 0;

  Capacity[1] := 8;      (* Establish jug capacities. *)
  Capacity[2] := 5;
  Capacity[3] := 3;

  Contents[1] := 8;      (* Initial contents of each jug. *)
  Contents[2] := 0;
  Contents[3] := 0;

  rewrite(Printer,':-2');

  (* Get things going by attempting to pour jug 1 into
   jar 1. Seems silly but it works. *)

  NextStep(1,1,Contents);

  close(Printer)
end.

```

VIP Writer I III

WORD PROCESSOR • SPELLING CHECKER • PRINT SPOOLER



SCREEN DISPLAY OPTIONS

VIP Writer III has a screen of 32, 40, 64, or 80 characters wide by 24 lines using the CoCo 3's hardware display with actual lower case letters. You can choose fore and background colors from up to 64 different hues. Color can be turned ON or OFF for the best possible display using a color or monochrome monitor or TV set. VIP Writer III has a built in on-line context sensitive help facility which displays command usage in easy to read colored windows. VIP Writer III also runs at double clock speed!

TEXT FILE STORAGE

There is a 49K text buffer and disk or cassette file linking allowing virtually unlimited text space. In addition VIP Writer III is compatible with the RGB Computer Systems HARD DISK.

“...Nearly every feature and option possible to implement on the Color Computer. The design of the program is excellent; the programming is flawless.” —The RAINBOW October 1983

EDITING FEATURES

VIP Writer III has a full featured screen editor which can be used to edit text with lines up to 240 characters long with or without automatic word wrap around. You can select type-over mode or insert mode. There is even an OOPS command to recall a cleared text buffer. Other editing features include: Type-ahead, typematic key repeat and key beep for flawless text entry, end of line bell, full four way cursor control with scrolling, top of textfile, bottom of textfile, page up, page down, top of screen, bottom of screen, beginning of line, end of line, left one word, right one word, DELETE character, to beginning or end of line, word to the left or right, or entire line, INSERT character or line, LOCATE and/or CHANGE or DELETE single or multiple occurrence using wildcards, BLOCK copy, move or delete with up to TEN simultaneous block manipulations, TAB key and programmable tab stops, three PROGRAMMABLE FUNCTIONS to perform tasks such as auto column creation and disk file linking for continuous printing.

TEXT FORMATTING

VIP Writer III automatically formats your text for you or allows you to format your text in any way you wish. You can change the top, bottom, left or right margin and page length. You can set your text flush left, center or flush right. You can turn right hand justification on or off. You can have headers, footers, page numbers and TWO auxiliary lines which can appear on odd, even or all pages. You can also select the line on which they appear! You can even change the line spacing! All of these parameters can be altered ANYWHERE within your text file.

TEXT FILE COMPATIBILITY

VIP Writer III creates ASCII text files which are compatible with all other VIP Programs as well as other programs which use ASCII file format. You can use VIP Writer III to create BASIC, assembly, PASCAL or C files. VIP Writer III also allows you to save and load files using DISK or CASSETTE in the case of an emergency. You can even read disk directories, display free space on a disk and rename or kill disk files.

SD ENTERPRISES

(503) 663-2865 8:30 AM to 5:00 PM PST
P. O. Box 1233 Gresham, OR 97030

PREVIEW PRINT WINDOW

The VIP Writer III features a paper saving format window which allows you to preview your document BEFORE PRINTING IT! You are able to see centered text, margins, page breaks, orphan lines etc. This feature makes hyphenation a snap!

PRINTING

VIP Writer III supports most any printer serial or parallel using the parallel interface described in Nov-Dec. '87 RAINBOW magazine, or an external serial to parallel interface, and gives you the ability to select baud rates from 110 to 9600. You are able to imbed printer control codes anywhere in your text file EVEN WITHIN JUSTIFIED TEXT! VIP Writer III also has twenty PROGRAMMABLE PRINTER SEQUENCES which allow you to easily control all of your printers capabilities such as underline, bold, italics, superscript and subscript using simple keystrokes. Additional printer features include: single sheet pause, print pause, word length and line feed selection.

PRINT SPOOLING

VIP Writer III incorporates a built in print spooler with a 49,000 character buffer which allows you to print one document WHILE you are editing another. You no longer have to wait until your printer is done printing before starting another job!

DOCUMENTATION

VIP Writer III is supplied with a 125 page instruction manual which includes a tutorial, glossary of terms and a complete index. The manual is well written and includes many examples to aid in understanding and application. VIP Writer III includes VIP Speller at NO ADDITIONAL COST. DISK \$79.95 Cassette version does not include VIP Speller. TAPE \$59.95

VIP Writer owners: Upgrade to the VIP Writer III Disk for \$49.95 or Tape for \$39.95. Send original product. Include \$3 shipping.

VIP WRITER - THE ORIGINAL

VIP Writer is also available for CoCo 1 and 2 owners and has all the features found in the VIP Writer III including VIP Speller except for the following: The screen display is 32, 51, 64 or 85 columns by 21 or 24 rows. Colors other than green, black or white are not supported. Help is not presented in colored windows. Double clock speed is not supported. Parallel printer interface is not supported. Print spooler is not available. Hard disk is not supported. Even so, the VIP Writer is a CoCo 1 or 2 owners best choice in word processors. VIP Writer includes VIP Speller at NO ADDITIONAL COST. DISK \$69.95 Cassette version does not include VIP Speller. TAPE \$49.95

VIP SPELLER - SPELLING CHECKER

VIP Speller works with ANY ASCII file created by most popular word processors. It automatically checks text files for words to be corrected, marked for special attention or even added to the dictionary. You can even view the misspelled word in context! VIP Speller comes with a specially edited 50,000 word dictionary, and words can be added to or deleted from the dictionary or you can create one of your own. DISK \$34.95

Please add \$3.00 for shipping and handling. COD orders add an additional \$2.25. Personal checks allow 3 weeks for delivery. All other orders are shipped the same day.

The 9th. Power

Randy Krippner

This month we're going to finish up PowerGraph by adding a routine that permits you to change any of the 16 colors in the palette to any of the 64 colors the Coco 3 can generate.

First load the procedures that make up PowerGraph. Edit the Powergraph procedure and change line 500 to this:

```
500 RUN chgpal \RETURN
```

* IMPORTANT * The character between "chgpal" and "RETURN" must be a BACKSLASH, generated by holding down CTRL and pressing the "/" key.

Begin a new procedure called chgpal, and type in the Basic09 listing that accompanies this month's article. After you've typed it in and saved it to disk (remember, save all of the procedures that make up PowerGraph into the same file by typing "save* PowerGraph" at the B: prompt), you're ready to go.

Use PowerGraph as described last time. Now, however, the Palette function on the menu is 'live'. When you click on Palette, the 16 colors that are in the current palette will appear on the screen. To change these colors, place the pointer on the color and click. Each time you click on a color, the color number will be incremented by 1 and the color will be changed.

Change all of the colors you wish. When you have the palette set as desired, move the pointer to the blank area between the last color on the menu and the double line that appears at the bottom of the window and click.

I didn't include a routine to save a palette to disk. If you wish, you can add a procedure to do this. The syscall in CHGPAL gets the current colors in the palette and puts them in the PAL array. Have your save procedure make this syscall to load the palette information into an 16 byte long array. Then open a disk file and store the contents of the array on disk. To restore a saved palette, open the file, read the values back in and use the GFX2 Palette function to set the palette registers.

Some people have been experiencing difficulty using the color bar and function menu, especially those using the keyboard mouse.

If you do not have the high resolution joystick interface, you must use the keyboard mouse. This is enabled by holding down CTRL and then pressing the CLEAR key. Pressing CTRL/CLEAR again toggles the keyboard mouse off. When using the keyboard mouse, the arrow keys move the mouse pointer on screen and F1 and F2 become the fire buttons. The left arrow will not backspace when the keyboard mouse is toggled on. You must use CTRL/H to backspace.

To move the pointer one pixel at a time, you must hold down SHIFT before pressing the arrow keys. Failing to hold down SHIFT will cause the pointer to move several pixels at a time.

You've no doubt noticed that when the menu or color bar windows pop up, the mouse pointer no longer responds if you try to move the pointer outside of the window. If the pointer is outside of the window when you call the menu, the pointer will not respond until you move the pointing device into the window area. Move the mouse or joystick all the way to the upper left corner of its range. This will get the pointer's coordinates into the proper area.

If you're using the keyboard mouse, this can be difficult. Move the pointer into the window area before you call up the menu. If you forget to do this, hold down the left arrow key for a second or two, then hold down the up arrow key. This should move the pointer coordinates into the window area so it will respond.

For the best results, you should really use the high resolution joystick interface and a mouse or joystick.

Also note that you must always re-select your drawing tool (point, line, box, etc.) after you change color, logic or use any of the non-drawing functions. Now let's take a look at the code itself.

If the only programming language you've used before is Extended BASIC, many of the things that go on in PowerGraph will be new to you. I wrote this version of PowerGraph to demonstrate as many of Basic09's control structures as I could. An experienced Basic09 or Pascal programmer would probably wince at the sight of this thing, but by writing it in this way I hoped to demonstrate how these structures are used in an actual program.

Just looking at examples in a book isn't the same as seeing these features in operation. If you examine the program, you should be able to determine how these features work. What you may find puzzling are the Syscall statement let's look at these more closely.

OS9 is made up of many small subroutines, and many of which may be accessed from Basic09 via a system call using Syscall. This is similar to using the USR statement of Extended BASIC to call a machine language subroutine in the computer's ROM. Syscall is more flexible because the OS9 system calls were designed to be used in this fashion, while the subroutines in ROM were not.

Take out your OS9 Level 2 manual and turn to the Technical Reference section. Chapter 8 gives descriptions of the various system calls that can be made to the OS9 operating system.

There are three types of OS9 system calls; User System Calls, Privileged System Mode Calls and System Calls. PowerGraph uses four User System Calls and several System Calls to perform functions that normally cannot be done with Basic09. The User System Calls are; GET STATUS, SET STATUS, READ (I\$READ) and WRITE (I\$WRITE). The User System Calls GET STATUS and SET STATUS themselves don't do anything. GET STATUS and SET STATUS are used to make System Calls. Which System Call is made depends on the function code loaded into the B register. The function codes used with GET STATUS and SET STATUS are listed under System Calls in the Technical Reference section.

Let's look at the CHGPAL procedure to see how Syscall is used. In order to display and change the colors in the palette, it's necessary to know just what colors are currently in each of the palette's 16 slots. There are different ways this could be done. We could, for example, define an array and constantly keep track of what's in the palette, but this is more work than is really necessary. The only place where this information is needed is in CHGPAL, and then only for a short time.

Basic09 can modify the colors stored in the palette, but it can't determine what colors are already there. But there is an OS9 System Call which can find this information. CHGPAL uses this system call like this:

First, a TYPE statement is used to set up a data type which mimics the 6809 micro-processor's internal registers. A variable, REGS, is declared, having the data type REGISTERS.

REGS will be the data structure we use to pass the required parameters to the OS9 subroutine called with Syscall.

The variable CALLCODE is used to hold the number of the User System Call we'll make. Every OS9 system call has a number which is used to identify it. Syscall looks at the number stored in CALLCODE and then makes the correct User System Call.

In this case, CALLCODE is set to \$8E. This is the SET STATUS call, listed on page 8-63 of the Technical Reference section. If you look at the documentation for SET STATUS, you'll note that there are two required parameters. Register A must be set to the path number and register B must be set to a function code. We're using the default I/O paths, so A is set to 0 (REGS.A = 0). But what about the function code?

As noted, GET STATUS and SET STATUS call still other subroutines. The actual routine that is called is determined by the function code stored in register B. These function codes are the System Calls, and are listed starting on page 8-112.

If you look at page 8-128 you'll find the System Call SS.Palet. This call reads the palette registers and stores them in a 16 byte buffer. The starting address of the buffer must be stored in the X register.

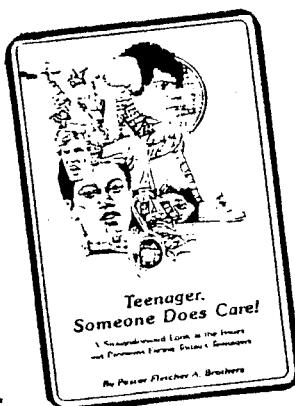
Before the Syscall is made in CHGPAL, REGS.B is set to \$91, and REGS.RX (RX is used instead of X because the variable X is already used elsewhere) is set to the address of our buffer, the array PAL(16), by the statement: REGS.RX = ADDR(pal).

After the Syscall executes, the buffer pointed to by the address in register X, the array PAL(), is loaded with the current values of the palette registers. Syscall is used throughout PowerGraph to enable the program to do things Basic09 cannot. The Diskfunctions procedure is a good example of this.

Finally

An answer for America's dying teens!

For your free copy of this dynamic and powerful book, call us today at
1-800-VICTORY.



Also;

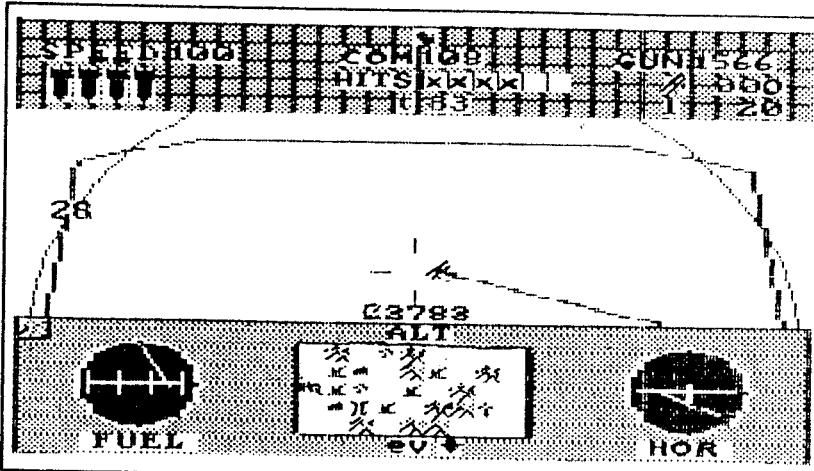
See and hear young people who have already found the answer at Freedom Village, USA in Lakemont, N.Y.

Check your local Christian radio and television stations for the Victory Today program.

For more information write
10 Freedom Village, USA Lakemont, NY 14857

Continued on page 29

MORE FROM ARK ROYAL!



WAR AT SEA: Wooden Ships simulate ship to ship battles during the 18th Century. Player controls a number of sailing ships from different nations and must pit his seamanship against the computer or another player.

NEW

A C E S: WWI Aerial Warfare (CC64K D HR ML)

RED ALERT: Star Ship Warfare (CC64K D HR MLS J)

WAR AT SEA: Wooden Ships (CC64K D HR MLS J)

Pro Football: Strategy Gridiron game (CC3 128K HR B) \$20

Okinawa: The Big Invasion (Screen Dump inc) (CC64K D HR ML) \$27

Blitzkrieg West: A Bigger Bulge (CC64K D HR ML) \$27

Bataan: Historical & Hypothetical games in one (CC64K D HR ML) \$29

Desert Fox: Rommel (CC64K D HR MLS) \$27

Task Force: Modern Naval War in the Med (CC64K D HR MLS J) \$27

D DAY: The 6th of June (CC64K HR ML) \$25

Battle Hymn: Battle of Gettysburg (CC64K D HR ML) \$25

Company Commander: Squad Level Wargame (CC32K SG MLS) \$25

(House to House Module included in Company Commander)

Additional Modules for Company Company 3.0

River Crossing \$17

Gemini \$17

Cauldron \$17

Beach Head \$17

Fire One! Submarine Simulation (CC3 D HR B) \$25

Fire & Steel: Waterloo Campaign (CC64K D HR MLS) \$22

Keyboard General: Bi-monthly newsletter yearly sub

Barbarossa, Luftflotte, Battle Hymn (256K) available Tandy 1000

New for the Tandy 1000

Gray Storm Rising: War in the North Atlantic

Codes: CC — Color Computer, all versions CC3 — CoCo 3 only

D — Disk only (no D means program available tape or disk)

HR — High Resolution SG — Semigraphics

ML — Machine Language

MLS — Machine Language Subroutines

B — Basic

J — Joystick

Write for free catalog!

Prices include shipping to USA and Canada. Others add \$3.00. COD's available in USA only, add \$3.50. Personal Checks accepted with no delays in USA. Others must send M.O. or Bank Draft in U.S. funds. Programs shipped within 24 hours except on weekends. Sorry, no bankcards. Color Computer and Tandy 1000, TM Tandy Corp.

Florida residents add 6% sales tax.

Canadians may order direct from: **M & M Software**, Post Office Box 1945, Slave Lake, Alberta T0E 2A0. Write M & M Software for information.

A C E S is a high resolution, completely machine language game of aerial warfare in WWI. Player flies on many missions to bomb enemy targets including airfields, enemy headquarters, anti-aircraft batteries, bridges and factories, but not player's own air base. He must dodge mountains and dogfight with the enemy's best, including, if unlucky, members of the dreaded Flying Circus. After he shoots down five planes he becomes an ACE and receives special consideration; but the game is far from finished. **A C E S** averages about 82 targets and over 100 enemy aircraft per game.

A C E S plays in real time and displays flight simulated dash and controls. Operates from the keyboard. Included in the display is a high resolution mini-screen featuring terrain, targets, and player's relative ground position. There are 8 zones in each map which changes as player flies over it. Game Save. (It could take days to win!) In addition, **NEWMAP** is included to allow for the creation of a zillion new maps. **A C E S** was created in part with AGS, developed by Ken Schunk. For all CoCo's.

RED ALERT: a starship combat simulator. Object of the game is to defeat the computer controlled enemy vessel by using your ship's capacities, strategic maneuvers, and your own smarts.

Luftflotte: Battle of Britain (CC32K SG MLS)	\$25
Stalingrad: The turning point. (CC64K HR ML)	\$25
Final Frontier: War in Space (CC32K D HR MLS)	\$25
Barbarossa: The War in Russia (CC64K HR ML)	\$22
RedStar: Nato vs Warsaw Pact (CC32K D HR ML)	\$22
DarkHorse: Redstar Sequel (CC64K D HR ML)	\$22
Midway: The Turning Point in the Pacific (CC32K HR MLS)	\$20
Escape From Denna: Dungeons! (CC32K SG MLS)	\$15
Tunis: War in the Desert (CC32K SG B)	\$15
Battle of the Bulge 1 or 2 player (CC32K SG B)	\$15
Phalanx: Alexander the Great (CC32K HR ML)	\$15
Rubicon II: Invasion game (CC32K SG B)	\$10
Guadalcanal: America Strikes Back (CC32K SG MLS)	\$10
Waterloo: Napoleon (CC32K SG MLS)	\$10
Bomber Command: Strategic Bombing Mission (CC32K SG MLS)	\$10
Kamikaze: Naval War in the Pacific (CC32K HR B)	\$10
Starblazer: Strategy Star Trek (CC32K SG MLS)	\$10
Mission Empire: Build an Empire in Space (CC32K SG B)	\$10
Galactic Taipan: Economics in Space (CC32K SG B)	\$10

 \$25

ARK ROYAL GAMES

P.O. Box 14806

Jacksonville, FL 32238

(904) 786-8603



CBASIC III EDITOR/COMPILER

The ULTIMATE Color Computer III BASIC COMPILER!!!

If you want to write fast efficient machine language programs and you don't want to spend the next few years trying to learn how to write them in Assembly language or with a cheap compiler, then CBASIC III is the answer!!!

CBASIC III is the only fully integrated Basic Compiler and Program Editing System available for the Color Computer 3. It will allow you to take full advantage of all the capabilities available in your CoCo-3 including 512K RAM, without having to spend years trying to learn assembly language programming. CBASIC III allows you to create, edit and convert programs from a language you are already familiar with Enhanced Disk Color Basic, into fast efficient machine language programs easily and quickly. CBASIC III supports all the enhanced hardware available in the CoCo-3, including Hi-Res Graphics, & Screen displays, Extended Memory and Interrupts (Keyboard, Timer, Serial & Clock). We even added advanced commands not available in Basic to give you a level of control only available to very advanced Machine Language Programmers. Plus we made it exceptionally easy to use, not like some other compilers. CBASIC III is the friendliest and easiest compiler available for the Color Computer III.

CBASIC III is a powerful tool for the Beginner as well as the Advanced Basic or Machine Language programmer. You can write programs without having to worry about the Stack, DP Register, memory allocations and so on, because CBASIC III will handle it for you automatically. For Advanced users, CBASIC III will let you control every aspect of your program, even generating machine code directly in a program easily.

CBASIC III features well over 150 Compiled Basic Commands and Functions that fully support Disk Sequential and Direct access files, Tape, Printer and Screen I/O. It supports ALL the High and Low Resolution Graphics, Sound, Play and String Operations available in Enhanced Color Basic, including Graphics H/GET, H/Put, H/Play and H/DRAW, all with 99.9% syntax compatibility. CBASIC III also supports the built in Serial I/O port with separate programmable printer & serial I/O baud rates. You can send and receive data with easy to use PRINT, INPUT, INKEY, GETCHAR and PUTCHAR commands.

CBASIC makes full use of the powerful and flexible GIMI chip in the Color Computer 3. It will fully utilize the 128K of RAM available and install 2 Ultra Fast Ramdisks if 512K is available, for program Creation, Editing and Compilation. You can easily access all 512K of memory in a Compiled program thru several extended memory commands that can access it in 32K or 8K blocks and single or double bytes.

CBASIC has its own completely integrated Basic Program Editor which allows you to load, edit or create programs for the compiler. It is a full featured editor designed specifically for writing Basic programs. It has block move and copy, program renumbering, automatic line number generation, screen editing, printer control and much more.

The documentation provided with CBASIC III is an 8 1/2 by 11 Spiral Bound book which contains approximately 120 pages of real information. We went to great lengths to provide a manual that is not only easy to use and understand, but complete and comprehensive enough for even the most sophisticated user.

CBASIC III is the most expensive Color Basic Compiler on the market, and well worth the investment. You can buy a less expensive compiler for your CoCo-3, and then find out how difficult it is to use, or how limited its features are. Then you'll wish you had bought CBASIC III in the first place. Dollar for dollar, CBASIC III gives you more than any other compiler available. If you can find a better CoCo-3 Basic Compiler then buy it!!!

Requires 128K & Disk \$149.00

DATAPACK III PLUS V1.1

SUPER SMART TERMINAL PROGRAM AUTOPilot & AUTO-LOG PROCESSORS X-MODEM DIRECT DISK FILE TRANSFER VT-100 & VT-52 TERMINAL EMULATION

- No lost data even at 2400 Baud on the COCO-3 Serial I/O port.
- 8 Display Formats, 32/40/64/80 columns at 192 or 225 Res.
- 50K Text Buffer when using the Hi-Res Text Display & Disk.
- ASCII & BINARY disk file transfer support via XMODEM.
- Directly record receive data to a disk file (Data Logging).
- VT-100 terminal emulation for VAX, UNIX and other systems.
- VT-100/52 cursor keys, position, insert/delete, PF & Alt. keys.
- Programmable Word Length, Parity, Stop Bits and baud rates.
- Complete Full and Half Duplex operation, with no garbled data.
- 9 Variable length, Programmable Macro Key buffers.
- Programmable Printer rates from 110 to 9600 baud.
- Send Files directly from the Buffer, Macro Keys or Disk.
- Display on Screen or Print the contents of the Buffer.
- Freeze Display & Review information On Line with no data loss.
- Built in Command Menu (Help) Display.
- Built in 2 Drive Ramdisk for 512K RAM support and much more.

Supports: R.S. Modem-Pak & Deluxe RS-232 Pak, even with Disk.

Requires 128K & Disk, \$59.95

EDT/ASM III

128/512K DISK EDITOR ASSEMBLER

EDT/ASM III is a Disk based co-resident Text Editor & Assembler. It is designed to take advantage of the new features available in the CoCo-3 with either 128K or 512K of memory. It has 8 display formats from 32/40/64/80 columns by 24 lines in 192 or 225 Resolution, so you use the best display mode whether you are using an RGB or Composite monitor or even a TV for your display. Plus you can select any foreground or background colors or even monochrome display modes. It will even support 512K by adding an automatic 2 drive Ultra Fast Ramdisk for lightning fast assembly of program source code larger than memory. There is also a free standing ML Debug Monitor, to help you debug your assembled programs. EDT/ASM III has the most powerful, easy to use Text Editor available in any Editor/Assembler package for the Color Computer.

- Supports Local and Global string search and/or replace.
- Full Screen line editing with immediate line update.
- Easy to use Single keystroke editing commands.
- Load & Save standard ASCII formatted file formats.
- Block Move & Copy, Insert, Delete, Overtype.
- Create and Edit files larger than memory.
- The Assembler portion of EDT/ASM III features include:
 - Supports the full 6809 instruction set & cross assembles 6800 code.
 - Supports Conditional IF/THEN/ELSE assembly.
 - Supports Disk Library file (include) up to 9 levels deep.
 - Supports standard Motorola assembler directives.
 - Allows multiple values for FCB & FDB directives (unlike R.S. EDT/ASM)
 - Allows assembly from the Editor Buffer, Disk or both.

Requires 128K & Disk \$59.95

TEXTPRO IV

The ADVANCED COCO-3 Word Processing System™

- 9 Hi-Res Displays from 58 to 212 columns by 24 lines in 225 Res.
- On Screen Display of Bold, Italic, Underline & Double Width print.
- Up to 8 Proportional Character Sets Supported with Justification.
- Up to 80 Programmable Function Keys & Loadable Function key sets.
- Fully Buffered keyboard accepts data even during disk access.
- Autoexecute Startup files for easy printer & system configuration.
- Pre-Defined Printer function commands & 10 Programmable ones.
- Supports Library files for unlimited printing & configurations.
- Disk file record access for Mail Merge & Boiler Plate printing.
- Completely Automatic Justification, Centering, Flush left & right.
- Change indents, margins, line length, etc. anytime in the text.
- Create and Edit files larger than memory, up to a full disk.
- Easily imbed any number of printer format and control codes.
- Built in Ultra Fast 2 drive RAMDISK for 512K support.

TEXTPRO IV is the most advanced word processing system available for the COCO-3, designed for speed, flexibility and extensive document processing. It is not like most of the other word processing programs available for the Color Computer. If you are looking for a simple word processor to write letters or other short documents, and never expect to use multiple fonts or proportional spacing, then most likely you'll be better off with one of the other simpler word processors. But, if you want a powerful word processor with extensive document formatting features to handle large documents, term papers, manuals, complex formatting problems and letter writing, then TEXTPRO IV is what you're looking for. It works in a totally different way than most word processing programs. It uses simple 2 character abbreviations of words or phrases for commands and formatting information that you imbed directly in your text. There are over 70 different formatting commands you can use without ever leaving the text you're working on. There are no time consuming, and often frustrating menu chases, you are in total control at all times. You can see what the formatted document will look like before a single word is ever printed on your printer. Including margins, headers, footers, page numbers, page breaks, column formatting, justification, and Bold, Italic, Underline, Double Width, Superscript and Subscript characters right on the screen.

TEXTPRO IV can even support LASER PRINTERS with proportional fonts, take a good look at this AD! It was done with TEXTPRO IV on an OKIDATA LASERLINE-6 laser printer!!! All the character sets used on this AD are proportional spaced characters, all centering, justification, and text printing was performed automatically by TEXTPRO IV.

Requires 128K & Disk \$89.95

HI-RES III Screen Commander

The DISPLAY you wanted but didn't get on your CoCo-3

- 54 Different Character Sizes available from 14 to 212 cpl.
- Bold, Italic, Underline, Subscript, Superscript and Plain character styles.
- Double Width, Double Height and Quad width characters.
- Scroll Protect form 1 to 23 lines on the screen.
- Mixed Text & Graphics in HSCREEN 3 mode.
- PRINT @ is available in all character sizes & styles.
- Programmable Automatic Key repeat for fast editing.
- Full Control Code Keyboard supported.
- Selectable Character & Background color.
- Uses only 4K of Extended (2nd 64K) or Basic RAM.
- Written in Ultra Fast Machine Language.

HI-RES III will improve the standard display capabilities of the Color Computer 3, even the 40 and 80 column displays have several features missing. For example, you can't use PRINT @ or have different character sizes on the same screen, even when mixing text and graphics with the HPRINT command. HI-RES III can give you the kind of display you always dreamed about having on your CoCo-3, with a wide variety of display options that you can easily use with your Basic or ML programs.

HI-RES III is totally compatible with Enhanced Color Basic and its operation is invisible to Basic. It simply replaces the normal screen display with an extremely versatile display package. With the full control code keyboard, you can control many of HI-RES III extended functions with just a couple of simple keystrokes.

Requires 128K Tape or Disk \$34.95

512K RAMDISK & MEMORY TESTER

RAMDISK is an ALL Machine Language program that will give you 2 ULTRA High Speed Ram Disks in your CoCo-3. It does not need or require the OS-9 operating system. It works with R.S. DOS V1.0 or V1.1 and it is completely compatible with Enhanced Color Disk Basic! Plus it allows your CoCo-3 to run at double speed all the time even for floppy disk access!!! It will not disappear when you press reset like some other ramdisk programs. The MEMORY tester is a fast ML program to test the 512K ram. It performs several bit tests as well as an address test so you know that your 512K of memory is working perfectly.

Requires 512K & Disk \$19.95

"The SOURCE III"

DISASSEMBLER & SOURCE CODE GENERATOR

The SOURCE III will allow you to easily Disassemble Color Computer machine language programs Directly from Disk and generate beautiful, Assembler compatible Source code.

- Automatic label generation and allows specifying FCB, FDB and FCC areas.
- Disassembles programs Directly from disk, unlike other disassemblers.
- Automatically locates Begin, End and Execution address.
- Output Disassembled listing with labels to the Printer, Screen or both.
- Generates Assembler source files directly to disk or printer.
- Built in Hex/Ascii dump/display to locate FCB, FCC & FDB areas.
- 8 Selectable Display formats 32/40/64/80 columns in 192 or 225 Res.
- Selectable Foreground & Background colors & Printer Baud rates.
- Built in Disk Directory an Kill file commands.
- Menu display with single key commands for smooth, Easy operation.
- Written in Ultra Fast Machine Language.

Requires 128K & Disk \$49.95

To order products by mail, send check or money order for the amount of purchase, plus \$3.00 for shipping & handling to the address below.
To Order by VISA, MASTERCARD or COD call us at (702) 452-0632
(Monday thru Saturday, 8am to 5pm PST)

CER-COMP LTD.

5566 Ricochet Avenue

Las Vegas, Nevada 89110

(702) 452-0632

Basic Help

Bill Bernico

Dear Bill,

I'm writing a program that uses DRAW strings to display letters and numbers on the graphics screen. Whenever I get to a certain point in the program, I get an FC error in 1300. I checked over line 1300 many times but it has no errors that I can find. What's happening? What am I doing wrong? My line 1300 reads:

```
1300 FOR X=1 TO LEN(A$):Y=ASC(MID$  
$(A$,X,1)):DRAW A$(Y):NEXT X:RETURN
```

George Schnicker
Redding, CA

Well, George, you don't give me much to go on. I assume you have first pre-defined all the necessary DRAW strings and that they are all error free. Your line 1300 may very well be correct, but if it tries to DRAW a string that is incorrectly defined, you will get an error in line 1300, not the line with the actual error. You may also be trying to DRAW something out of the bounds of your graphics screen. A DRAW string may inadvertently contain a command to DRAW R350 when you actually wanted R35. Look over all your DRAW strings first. Chances are your error is in one of them.

The following is a condensed version of the type of program you are talking about. This sample program only contains 10 DRAW strings, but you could actually include DRAW strings 32 through 122 to include all the letters, number and characters of the keyboard.

Send your programming
questions to Bill
Bernico's BASIC Help

c/o CoCo Clipboard
3742 U.S. 20, Box 3
Fredonia, N.Y. 14063

```
10 DIM A$(115):'DIMENSION A$ TO  
THE HIGEST NUMBER OF STRINGS  
USED.  
20 PMODE4,1:SCREEN1,1:PCLS1:COLO  
R0,1:'SET YOUR GRAPHICS SCREEN  
PARAMETERS  
30 'DEFINE YOUR DRAW STRINGS  
40 A$(32)="BR6":'SPACE  
50 A$(67)="BR5NFL4GD6FR4EBU7BR3":  
'UPPERCASE C  
60 A$(97)="BD3R3FD3GL3HUER4BU5BR  
3":'LOWERCASE A  
70 A$(98)="D8R4EU3HL4U3BR8":'LOW  
ERCASE B  
80 A$(100)="BR5D8L4HU3ER4U3BR3":  
'LOWERCASE D  
90 A$(105)="BD8R4L2U5NLBU2NUBUBR  
5":'LOWERCASE I  
100 A$(108)="ND8BR4":'LOWERCASE  
L  
110 A$(111)="BD7NU3FR3EU3HL3BU3B  
R7":'LOWERCASE O  
120 A$(112)="BD3ND8R4FD3GL4BU8BR  
8":'LOWERCASE P  
130 A$(114)="BD3D5U2E3RFBU4BR3":  
'LOWERCASE R  
140 A$="CoCo Clipboard":'DEFINE  
THE CONTENTS OF A$  
150 DRAW"BM50,90":'DEFINE THE AR  
EA TO BE DRAWN  
160 GOSUB190:'GOSUB'S TO THE WOR  
KING LINE  
170 GOTO 170:'THIS LINE IS ONLY  
HERE TO KEEP THE IMAGE ON  
THE GRAPHICS SCREEN  
180 THE NEXT LINE HAS THE JOB  
OF CHECKING WHAT YOU'VE PUT  
INTO A$, FINDING ITS ASCII  
VALUE, AND DRAWING THAT  
PARTICULAR CHARACTER  
190 FORX=1TOLEN(A$):Y=ASC(MID$(A  
$,X,1)):DRAWA$(Y):NEXT:RETURN
```

A Better Display

Jim DeStafeno

Okay, anyone with a CoCo-3 and a poor 80 column display raise your hand. (There's lots here for you CoCo-2 men as well. Don't miss the end of the article. You Tape-Forever men might find some good stuff in the programming tips, and you Color Viewing fellows may find an improved display too.)

Now, everyone that hates the DIR display and its lack of FREE granule notation, raise your hands. Lastly, all you guys with monochrome who are tired of typing in a litany everytime you fire up, raise your hand.

I don't know about you, but I am glad there isn't another question; I have (had) both hands and a leg in the air. The following cares for all three with just a few key strokes.

"COLORset" LOADs setting the proper PALETTE and ATTR values and goes directly to the 80 column display. It then asks if program LOADING or Exiting is wanted. Choosing LOAD results in a "what DRIVE" question. The answer results in a one screen display of every program on the disk - what a first time sight - and the number of FREE granules too.

Then using the arrow keys and the highlighting capabilities of the -3, the program to LOAD is selected. Note: Selecting a position with no program will clear all and bring up the first menu. The ESCape/BREAK key ENDs the program; clears the screen and displays the OK sign from any place in the program.

When the desired program to LOAD has been high lighted with the arrow keys, it is selected with the SHIFT/UP ARROW keys. The ENTER key wasn't used because it would be too easy to press unintentionally. The selection brings up the last screen. It asks for a choosing of a 32, 40 or 80 column screen display. The Exit choice comes directly to this screen.

This last choice LOADs and RUNs the selected program in the selected characters /line mode. However, if you can't see the LOADED program, the whole exercise is somewhat pointless. Enter the reason for Listing 1, "COLORtst".

Its been reported to me different monochrome monitors display better with different

CLS and ATTR values. Color Test is a quickie that displays all possible CLS_n and ATTR_{nX,nY} values, while the PALETTE values are set to black background and white letters. The most legible CLS and ATTR values can be chosen and used in COLORset, the real program in this article; as well as all your own programs.

"COLORtst" is a one time shot and I expect you, like me, hate to type in instructions. To minimize the typing, each screen begins with a roman numeral. The corresponding explanation paragraph below begins with the same roman numeral, as in:

(I) Just follow the displayed instructions. This screen tests all the values for CLS_n, CLS1 thru CLS8. What you are looking for is a screen, border and all, filled with one solid color; no lines at all. My monitor displays a "solid" color only with 1 and 5, black and yellow, amber monitor. All the other numbers have a display with coarse lines. Write down the number(s) that yield the "solid" display(s). From here on I'll refer to these "solid" numbers as the CLS Number.

Note that after all eight CLS Numbers are checked you are asked if you want to do it again. It is important that you are sure of your number(s).

(II) This screen asks for your first CLS Number to be inputed. If more than one CLS Number was discovered, the others can be tested later. I tested 1 first and 5 later.

(III) Here you will determine three sets of values for the ATTR command. The top of the screen displays eight side by side vertical "bars" of equal width with X and Y values noted in them. Think of the bars as being numbered 1 thru 8, left to right. Find the most legible X/Y set in the bar with the same number as the CLS Number you just inputed.

My first CLS Number test was 1, so I searched the left hand bar and found X=0 Y=0 to be the most legible. The values you choose here are what I'll refer to as your Primary ATTR Number. Write the number set down and label it "Primary".

Next determine your Shaded ATTR Number by choosing a X/Y set in a "bar" with lines, and with text that you can read easily. Write yours down. Mine are X=0 Y=6.

Finally determine your Contrasting ATTR Number. Find a solid, or near solid bar, not either of the two bars you already used. Find the most legible X/Y set in this chosen bar. Mine is X=4 Y=4. This Contrasting ATTR Number should be used sparingly for high lighting, etc.

You now have four labeled values; a CLS Number, Primary ATTR Number, Shaded ATTR Number and Contrasting ATTR Number. If your numbers don't agree with mine, find my values in "COLORset" and substitute your values in their place.

The program continues by asking if you want to do it again, meaning do you want to input a second CLS Number. When you are finished with this test, you are finished with the program.

CODING TIPS: I don't mean to give the impression that I think these programs are perfect, but the following works well and seldom, if ever, have I seen them in published programs. With that in mind use what you feel to be of value.

You've already seen two. Note the program names. Use upper and lowercase letters to separate words in the name, not spaces or dashes, it is easier to read. To shorten words, omit vowels, working from right to left. Using this thought, COLOR SET became COLORset, not COLORSET; while COLOR TEST became COLORtst.

However this can cause a trap. If the name end in lowercase and the computer isn't switched back to uppercase and if the program commands are in uppercase the program will not react to the inputs. That is the case with this program. The solution is to either remember to switch the "case" or put in the case POKEs. I haven't put the POKEs in this program, but if you want, the uppercase, POKE282,1 would go into line 10 and POKE282,0 in line 800.

The keying of program menu screens to the documentation helps the user, and the writer, from getting confused. I've used the roman numerals here.

INKEY\$ - I've seen programs that cause the user to pound the keyboard to get the computer's attention. That is normally due to INKEY\$ not being invoked when the key is pressed. The normal coding of INKEY\$ causes these missed key strokes. There is a way to use it so that it never misses, precede it with EXEC44539. It causes the CPU to wait for a key stroke. So if INKEY\$ comes after the EXEC, a key stroke can't be missed.

Check out COLORtst, lines 30 and 110. Note that this use allows INKEY\$ to be used any place in a line and uses less memory. However, it can only be used in this way when there are only two choices, and one of them can be a default; such as a Yes / No situation. To be correct, the choice label in the program should read (Y / Any Key), not (Y / N).

In COLORset, lines 40 with 20 and 30, 110 with 100 and line 510 with 500 are a bit more

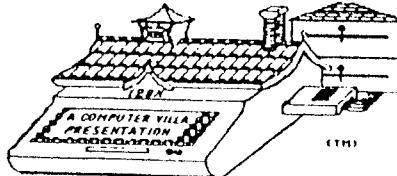
heavy duty. 510 would even be faster if the STRING\$ portion had been set to a STRING\$ value in the beginning of the program, such as A\$, and A\$ had been used in the INSTR command. This is a v-e-r-y powerful and fast combination of commands.

This command set not only cares for the desired action, but it is also user error proof; hitting an unidentified key lets the user try again, note the +1 and the first line number after the GOTO.

Maybe the 40 / 20, 30 lines could have been coded using the normal method, but if the suggested program enhancements listed below are implemented, this line-set will have five or more choices to contend with. Lines 110 / 100 are a bit of over kill, but there will be no missed keyboard inputs.

The POKEs at the beginning of line 510 cause the "key repeat" during the program to LOAD the selection. They change the sensed state of the up and down arrow keys to "released", which causes their table to reset itself. Then since you have the key down it resets itself again to the waiting state just in time for the program to come back and INKEY\$ to react to the depressed key. 343 and 344 work for the left and right arrows, but it is too fast to invoke for just four positions.

Continued on page 16



Magnavox
8CM515
analog rgb monitor w/cable

\$310.00
DELIVERED

Switch selectable RGB Color, Composite (for PMODE 4) and green text screen. RGB cable included in the price. Also watch your VCR on this monitor for fantastic color!

Certified Check or Money Order only.

Computer Villa
Terry Simons
(author of Home Pack & Fast Check)
1328 48th St.

Des Moines, Iowa 50311
(515) 279-2576 till 11pm Central

Continued from page 15

The arrow key driven selection routine is simple AND prints in column order, as we normally read, rather than row order as is normally programmed. Of course it wouldn't work with a true terminal. That would require more complex code, but few of us, if any, are using RS BASIC and a terminal.

As should be done, GOSUB/RETURN is used only when the GOSUB line number is used more than once from different parts of the program. GOTO is used in all other places.

Of course using the new text blink command and three shades of "color" without having to resort to graphics was fun.

ENHANCEMENTS to "COLORtst" would be a waste of time, save one. A looped value substitution in PALETTE_n would be interesting.

However, "COLORset" would benefit with SOUND added, plus ML routines for key repeat and speeding the disk drive action; shorter Up To Speed Checking and 6ms Track To Track movement.

As the program is written, it is meant to be put on all your disks and LOADED with the DOS command. As such the LOAD/Exit question menu should be deleted. However, the program could be made into a real disk manager by adding COPY, KILL, RENAME and MERGE, selected at this menu. If one wanted to go heavy duty, disk examine and change could be added, along with a granule map.

While I am writing a program, I like to make SAVES at each mile stone. Check the last lines in each program. I just type in GOTO1000 and can make either a single or double SAVE.

Of course the program could be converted to be used with the CoCo-2. Displaying the directory would be a little tricky, only two columns could be displayed and therefore only 32 program names on one screen. To display and select all possible 68 programs, the display would have to scroll up and down using the arrow keys. I expect Ted would print any of these enhancements if someone sent in the code.

To put the icing on the cake, set up "COLORset" to LOAD and RUN via the DOS command.

(HELP! - Does anyone know of a POKE, etc. that will change the timing of the normal 80 column screen write so the space between the lines and characters will be larger? My screen is not filled using the normal timing. If you don't have the code for using DOS, send a SASE, enclose a note telling me what it is for and I'll send you the code.)

I already have this program working on the CoCo-2 with the Disto 80 column card. If you have Disto's 80 columns, I'll send the converted "COLORset" Listing or on a Disk for

that 80 column set, on a "Freeware" basis; send what you think it is worth. Keep in mind the price of a mailer, postage, a disk and my time. The new address is RD 1, Box 375 Wyoming, DE. 19934.

Happy computing for now.

Editors Note: This is a 40 column listing which will let you take advantage of the CoCo 3.

LISTING 1, "COLORtst", Color Test:

```
0'*****  
1'*+++ "COLORtst" +++ *  
2'* DISCOVERS BEST BACK- *  
3'* GROUND, FOREGROUND AND *  
4'* SHADED 'COLOR' FOR A *  
5'* MONOCHROME/COMPOSITE *  
6'* COCO-3 SETUP *  
7'*****  
8'  
10CLS:PRINT:PRINT"(I) PRESS ANY KEY REPEATEDLY AND NOTE NUMBER DISPLAYED EACH TIME. THEY WILL BE 1-8. SOME MAY NOT SHOW.":PRINT:PRINT" REMEMBER THE NUMBER(S) WITH THE MOST SOILD -FULL SCREEN"  
20PRINT" DISPLAY. MINE IS 1. PRESS ANY KEY NOW.":EXEC44539  
30WIDTH80:PALETTE0,0:PALETTE8,63:FOR C=1 TO 8:CLSC:PRINT"--- "C" ---":EXEC44539  
:NEXTC:ATTR0,0:PRINT:PRINT" DO IT AGAIN ? (YES / NO)":EXEC44539:IF INKEY$="Y" THEN 30  
40PRINT:PRINT"(II) INPUT THE 'SOLID' DISPLAY NUMBER ? ";:EXEC44539:C=VAL(INKEY$):CLSC  
50FOR X=0 TO 7  
60FOR Y=0 TO 7  
70ATTRX,Y  
80PRINT"X="X;"Y="Y; 90NEXTY:NEXTX:ATTR0,0  
100PRINT:PRINT:PRINT"(III) PICK OUT THE MOST LEGIBLE X/Y NUMBERS IN YOUR 'BAR'. NOW FIND THE MOST LEGIBLE X/Y NUMBERS ON A BACKGROUND WITH LINES. FINALLY PICK OUT AN X/Y SET ON A CONTRASTING 'SOLID' BACKGROUND.  
110PRINT:PRINT"DO YOU WANT TO PUT IN AN OTHER 'SOLID' DISPLAY NUMBER ? (YES / NO) ";:EXEC44539:IF INKEY$="Y" THEN PRINT:GOTO40 ELSE CLS:END  
120STOP  
1000SAVE"COLORtst":0":PRINT"ok":EXEC44539  
1010SAVE"COLORtst":1":STOP
```

LISTING 2, "COLORset", COLOR SET:

```
0'*****  
1'*+++ "COLORset" +++ *  
2'* SETS 'WIDTH' & 'COLOR' *  
3'* FOR A MONOCHROME/ *  
4'* COMPOSITE/COCO-3, PLUS *  
5'* PROGRAM LOADER *
```

Continued on page 32

Database Tutorial Pt. II

Rush Caley

In last month's column, I made one rather hasty comment. I said that the database we discussed last time would not be "all that difficult to design". After having submitted Part I to the magazine, I discovered a problem that required a massive database change from my original design. This illustrates my point concerning the importance of knowing PRECISELY what your requirements are ahead of time. The function of a database—or even the reason for its existence—is to support REPORTS. These can be on screen inquiries or hard copy reports. So the first step in building a database is examination of the requirements and the determination that your "GIVENS" will provide adequate information to support them.

The "GIVENS" in this situation are these:

1. We have an assembly line type of process in which the employees will fill containers of different sizes with assorted foodstuffs. The purpose being to fill as many containers as possible during a shift.

2. There will be more than one shift.

3. EACH shift MAY be required to fill more than one size container and MAY need to change the type of foodstuff. Each of these situations have different established rates of speed expected to be met.

4. On most shifts, DOWN TIME occurs.

5. DOWN TIME is due to DEPARTMENTAL, MECHANICAL, or ELECTRICAL problems. (These three for our purposes.)

6. GIVEN DEFINITIONS:

A. ACTUAL UNITS - # of CASES of goodies filled in a shift.

B. STANDARD SPEED - A number of cases per minute EXPECTED to be filled during a "run" for a specific size container. Each size container being filled with a specific foodstuff has a pre-determined # of cases per "run" expected. This is set by management and is derived by calculations for which we have no information.

C. ACTUAL RUN TIME - How many actual minutes elapsed to complete the number of cases filled during the run.

D. STANDARD RUN TIME - This is the result of a calculation. It is derived by dividing the ACTUAL UNITS by the STANDARD SPEED. So if you completed 3129 cases during a 450 minute shift; and the STANDARD SPEED was 7.5 cases per minute, then your STANDARD RUN TIME would be $3129 / 7.5$ or 417.20 minutes.

E. DOWN TIME - According to the supplied information, the DOWN TIME would be the difference between the ACTUAL TIME and the STANDARD RUN TIME. Using the figures from above, the DOWN TIME on that particular run would have been 32.80 minutes.

F. EFFICIENCY - The formula supplied to describe how well the shift performed in relation to its defined goals established by the STANDARD SPEED. It is the STANDARD RUN TIME divided by the ACTUAL RUN TIME expressed in a percentage. Or $\text{EFFICIENCY} = \text{STANDARD RUN TIME} / \text{ACTUAL RUN TIME} * 100$. In the above example, $417.20 / 450 * 100 = 92.91\%$

Most of the data fields to be used in our database can be inferred from the given information above. Others I have added for different reasons. A complete list of all data fields and their definitions are listed in a separate section for your convenience. I will assume that you will familiarize yourself with the list so that later references to fields will not be confusing. Remember that the data fields will comprise the "Y" axis of our database. In order to refresh your memory on my mental picture of the database, you might take another look at last issue where "X" and "Y" axes were discussed.

Each record in the file has to have a "name". The record names comprise the "X" axis. They can be names of people, account numbers, check numbers, etc. When we set up the file, we need to have some standard for naming records—or a record naming convention. A little creativity in this area goes a long way toward efficiency in record selection after the file is set up. There are two reasons to use some imagination here. First, the record "names" will be held in memory. This allows instantaneous access to one or more records based on selection. Secondly, all selection criteria applies to the "X" axis as well as to be "Y" axis. So you have the ability to access based on ">", "<", "=" "Range" or "Generic" selections to record names held in memory at a blinding rate of speed.

Listed below are record names entered for one shift for one day! These are names only and have no data. Take a close look and I will then explain the reasoning behind this convention. Each record name implies a downtime entry for some particular reason. Incidentally, in WORKBASE, we are limited to 12 characters for the record name.

When entries for the shift have been completed, or if there has been a change in product with a different STD SPEED, the operator must enter a "GT" type record for all previous records to store totals for that shift or partial shift.

X "	880104-1A-A	field1	field2	field3	field4	field5	etc. "
	880104-1A-B						
	880104-1A-C						
AXIS	880104-1A-D	<<<<<< " Y " AXIS >>>>>>>					
	880104-1A-E						
	880104-1A-F						
	880104-1A-GT						
	880104-1B-A						
	880104-1B-B						
	880104-1B-GT						

The first 6 characters represent the date. Although there will be a DATE field in the file, this makes selection by date very fast. Also, when the file is sorted based on record name, the records are always in order of date because the numbers are in a YY/MM/DD format. The next 2 characters after the "--" delimiter says 2 things. First, it says that the shift in this record is shift #1. The second character, "A" or "B" represents the fact that although the same shift is represented, there has been a change in product line and a different STANDARD SPEED is in effect. The last character or characters represent ascending order of items being entered. The "GT" means that certain totals will be calculated for all records that precede it.

In the above example, SHIFT #1 had 6 delays (A-F) during one portion of the shift. And after changing products and calculations being based on a different STANDARD SPEED, they had 2 delays.

I can now access all of these records by selecting record name "880104". I can select only SHIFT#1 records by selecting "-1". I can select only part of SHIFT#1 by selecting "-1A" or "-1B". I can select only the totals for each part of the shift by selecting "-GT". These selections require no disk access and happen instantly as record names are in memory. Nifty, eh? Like I said, you must have a record name, and you might as well make it work for you as not.

So now let's look at a typical entry for each type of record. The data entry operator will have the completed forms handed in by the shift supervisors. For each indication of downtime during the shift, he will enter one record with the information on the form.

880104-1A-A							
DATE:	01/04/88						
SHIFT #:	1						
PRODUCT:	24 OZ. SYRUP						
DWNTIME TYPE	MECHANICAL						
REMARKS:	CAPPER: ADJUST CHUTE						
	HEIGHT & AIR PRESSURE						
MECH-CTR:	1						
DWNTIME:	5						
SHIFT START:	07:00 AM						
SHIFT END:	03:30 PM						

880104-1A-GT

DATE:	01/04/88					
SHIFT #:	1					
PRODUCT:	24 OZ. SYRUP					
STD SPEED:	7.50					
ACT. UNITS	3129					
ACT RUN TIME:	450					

And that's it! STD RUN TIME, DOWN TIME, AND EFFICIENCY for each "GT" record will be calculated by a WORKBASE calculation procedure and stored in those fields. In the majority of cases, there will only be one "GT" record per shift. A second one is only required when a change in PRODUCT has a different STD SPEED than the previous one.

I might add, that there are two features in WORKBASE that make all of this very simple. First, as records are added, they can be copied from the previous record if one chooses. In this way, fields that do not change value, do not have to be re-keyed for each record. Secondly, and most important. The calculation procedures and report procedures are stored on the disk. So the user does not have to re-key macros for each time he calculates or wishes a report. All he need do is open the file and execute the procedure.

Now is the time I would like to discuss the reports I have designed to fulfill the request of the user who wrote with the original problem. In the first issue I ended by saying that we would examine the data from three vantage points. One report will examine the data by SHIFT # with information sorted in such a way that sub-totals are reported by SHIFT #. The second report will look at the same data, but will be sorted, subtotalled and viewed by PRODUCT. And finally, the third report will examine the identical information from the standpoint of the DWNTIME TYPE. When you see these reports, you will be able to see that although the information is similar, seeing it from three separate vantage points allows for more complete analysis of what is happening. Specific problems in specific areas can be isolated and hopefully solved to attain more efficiency.

Continued on page 29

ALL NEW! CCMDisk Service

PRESENTS

ClipDisk

FULL YEAR (6 issues) Only \$49.95 !

Now all the BASIC, PASCAL, C, BASIC OS9, Assembly Language and OS9 programs you see in CoCo Clipboard magazine are available on disk. With ClipDisk you'll save time and be more productive computing with your Color Computer.

A subscription to ClipDisk is \$49.95 for 6 issues - just like CoCo Clipboard Magazine. And single issues are available as well for \$9.95. Of course in order to get the full benefit of ClipDisk you'll need the corresponding issue of CoCo Clipboard. ClipDisk is not a stand alone item, but rather a real help in enjoying Clipboard and your CoCo.

So don't delay, get your subscription in today! CCMDisk Service will begin shipping your ClipDisk starting on April 1, 1988.

To: CCMDisk Service
RD # 1
Box 375
Wyoming, Delaware 19934

Make your check payable to:

ClipDisk

I've enclosed (check one) 1 year of ClipDisk \$49.95
 single copy 9.95

Specify month desired _____

Canadian customers can also obtain ClipDisk for \$49.95 when paid in U.S. Funds.

Name _____

Address _____

State _____ Zip _____

Please allow 4 to 6 weeks for delivery.

ADS?

ATTENTION! COCO SOFTWARE
AND HARDWARE VENDORS



ARE ADVERTISING RATES
MAKING YOUR HAIR STAND ON END?

WE HAVE AD RATES YOU CAN AFFORD

Let us help you to market your
product without having
to pay an arm and a leg.

For Information write or
call our office
at (716)679-0126

Monday-Friday 9am-2pm

Saturday 9am-6pm

**CoCo
Clipboard Magazine**

3742 U.S. 20 Box 3,

Fredonia, N.Y. 14063

On Line Conference on CompuServe

Join Ted Paul from CoCo Clipboard
and Dan Robins of Computer Shopper
each month for an nationwide on
line conference on CompuServe.

It's easy and it's fun and you'll
have the chance to meet and talk
to our special guests - the
programmers and hardware
developers who are making the CoCo
grow!

The conferences are in the CoCo
Forum on CompuServe and start at 9
pm Eastern on the second Saturday
of the month. The next three dates
are:

March 12 th.
April 9 th.
May 14 th

Special guests and topics T/B/A
Plus you'll have the chance to
discover all the fun of using
CompuServe to meet new CoCo users
from across the U.S. & Canada and
exchange information, comments and
suggestions.

Remember - the second Saturday of
each month at 9pm Eastern. Just
type "Go CoCo" and follow the
menu.

Continued from page 2

have conferences too, I don't care. Our conferences are monthly, our conferences will have special guest speakers and our conferences are on CompuServe so when you're done talking you can do a heck of a lot more on line than any place else. I hope the CompuServe people see this (they will, I've got the clasp envelopes all made out with their names on them) and realize the action for the CoCo is here in Clipboard and in the CoCo Forum.

Finally (I've got to wrap this up) we are announcing a price increase for subscriptions starting in August. However we're going to make it as painless as possible. Now for those of you who don't have the common sense God gave cabbage (thanks Momma) and who can't appreciate one of those Motel 6 commercials by

Tom Bodette, read no further. But for those of you who get a bang out of Tom Bodette and grin wide when Momma let's a zinger fly, we've got a deal for you. You can subscribe or renew for up to two years and beat the price increase. If you take the two year subscription we'll even knock off a buck to boot. Here's the deal:

Orders postmarked before August 1, 1988

1 year of Clipboard (6 issues) --- \$12.00
2 years of Clipboard (12 issues) - \$23.00

Orders postmarked AFTER August 1, 1988

1 year of Clipboard (6 issues) --- \$15.00
2 years of Clipboard (12 issues) - \$28.00

Canadian subscriptions will now be \$18.00 per year and other foreign subscriptions will be \$27.00

That's it for now I, oops almost forgot! We are announcing CCMDisk Service. Starting March 1, 1988 you can order Clipboard on disk. All the programs in the magazine on disk and the ad in this issue will tell you how. Be sure to send your order to the proper address in Delaware for CCMDisk Service. And we're also as pleased as punch to welcome Clearbrook Software Group's newsletter to our pages. The CSG users around the world will be seeing Clipboard from now on. Glad to have you fine folks on board. Well I'm done, the article is finished (AMEN) enjoy, and keep in touch.

For the Coco User who has Everything . . .

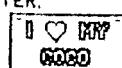
MOUSE COVER HIT - CONTAINS EVERYTHING YOU NEED TO MAKE A COVER FOR YOUR MOST IMPORTANT GRAPHICS TOOL. EASY INSTRUCTIONS AND QUICK ASSEMBLY. YOU'VE SEEN THIS MOUSE BEFORE!

\$3.50/HIT



COCO LICENSE PLATE - SHOW EVERYONE HOW YOU FEEL ABOUT YOUR FAVORITE COMPUTER. DRESS UP YOUR CAR OR COMPUTER ROOM.

\$2.95



COCO IN STITCH - EIGHT CROSS STITCH PATTERNS WITH THREE FONTS. EACH PATTERN IS COCO RELATED AND CAN BE STITCHED ON ANY COUNT CLOTH.

\$3.95



COME SEE US AT RAINBOWFEST CHICAGO!

Specialty Projects

4810 McCrory Ave.

Memphis, TN 38122

901-682-8737

ADD \$3.00 FOR SHIPPING AND HANDLING.
CASH, CHECK, OR MONEY ORDERS ONLY!
PLEASE, NO CODS OR CREDIT CARDS.

EVEN IF YOU DON'T HAVE A

HARD DISK

YOU CAN STILL SUPERCHARGE YOUR COCO 1,2, OR 3 WITH

RGB-DOS(HD)

HERE ARE JUST SOME OF THE FEATURES OF RGB-DOS(HD):

- FULLY COMPATIBLE WITH RS-DOS
- WILL RUN TWO HARD DISK DRIVES
- WILL AUTO-EXECUTE ANY PROGRAM
- FULL SCREEN DIRECTORY DISPLAY
- ELECTRONIC DISK LABELING
- IMPROVED "COPY" COMMAND
- "RUNM" COMMAND FOR M/L PROGRAMS
- ...AND MUCH MUCH MORE!

RGB-DOS(HD) COMBINES ALL THESE FEATURES WITH THE ABILITY TO RUN ANY SIZE HARD DISK DRIVE IN BASIC!

DO YOUR COCO A FAVOR...SUPERCHARGE IT WITH RGB-DOS!

28 PIN ROM WITH USERS MANUAL... \$74.95

ALSO ASK ABOUT OUR PC/XT-TO-COCO KEYBOARD ADAPTER AND OUR FULLY SCSI COMPATIBLE HARD DISK INTERFACE



COMPUTER
SYSTEMS

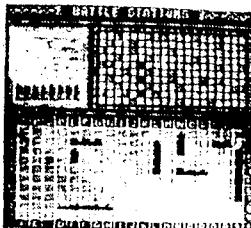
294 STILLWELL AVE
KENMORE, NY 14217

(716) 876-7538

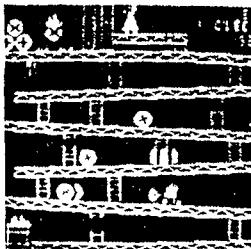
HOT PICKS IN '88

GAMES

Vegas Slots.....	\$28.95
Video Cards & Keno.....	\$28.95
Wizard's Den.....	\$20.95
Moneyopoly.....	\$19.95
Vegas Game Pak.....	\$24.95
LunchTime.....	\$18.95
Battle Stations.....	\$19.95
The Misadventures of Eddie.....	\$19.95
Buzzard Bait.....	\$19.95
The Frog.....	\$19.95
Crash.....	\$14.95
Quix.....	\$19.95
Elec*TRON*.....	\$19.95
The Sailor Man.....	\$24.95
Ms. Maze.....	\$19.95
Katerpillar II.....	\$19.95
Trapfall.....	\$19.95
The King.....	\$24.95
Flight 16.....	\$29.95
Flight areas & airports.....	\$14.95
P51 Mustang/ Flight Simulator.....	\$29.95
Worlds Of Flight.....	\$29.95
Approach Control Simulation.....	\$24.95
Four Cube.....	\$14.95
Maui Vice.....	\$19.95
Cuber.....	\$19.95
The Martian Crypt.....	\$19.95
Black Beard's Island.....	\$14.95
TV Blackout Bingo.....	\$19.95
Iron Forest.....	\$27.95
Gantelet II.....	\$27.95
Grand Prix Challenge.....	\$27.95
Gantelet.....	\$27.95
Wrestle Maniac.....	\$27.95
Mission F-16 Assault.....	\$27.95
Karate.....	\$27.95
Knock Out.....	\$27.95
Marble Maze.....	\$27.95
Paper Route.....	\$27.95



BATTLE STATIONS



THE KING

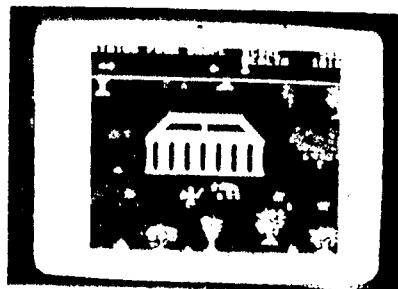
EDUCATIONAL SOFTWARE

Count To 100	\$28.95-C	\$31.95-D
Money Math	\$28.95-C	\$31.95-D
Primary Number Ski	\$28.95-C	\$31.95-D
Subtraction	\$28.95-C	\$31.95-D
Alphabet	\$28.95-C	\$31.95-D
Number Readiness	\$23.95-C	\$26.95-D
Before/ Between/ A	\$28.95-C	\$31.95-D
Clock Arithmetic	\$28.95-C	\$31.95-D
Addition	\$28.95-C	\$31.95-D
Number & Color Wor	\$28.95-C	\$31.95-D
Multiplication Duo	\$20.95-C or D	
Story Problems	\$21.95-C or D	
Fractions	\$34.95-C or D	
Fact Pack	\$28.95-C or D	
Pre Algebra I Inte	\$32.95-C or D	
Pre Algebra II	\$32.95-C or D	
Teachers' Database	\$41.95-C or D	
Teachers' DB 2	\$58.95-DISK ONLY	
Vocabulary Managem	\$41.95-C or D	

MISCELLANEOUS SOFTWARE

Autoterm	\$28.95-C	\$34.95-D
Code Practice	\$ 9.95-DISK	
VCR File	\$19.95-DISK	
Super Disk Util-3	\$19.95-DISK	

Please allow 2 - 3
weeks for
delivery



WIZARD'S DEN
\$20.95 DISK ONLY

SUN PRODUCTS

5455 Hansel Avenue
Bldg. L · Suite 7
Edgewood, FL 32809-3405
(305) 240-4784

Send \$3.00 for current catalog.
We accept cash, check, MO, or COD

Offering Products & Support for the Tandy/ Radio Shack Color Computer 1, 2, and 3

Product Reviews

Paul Bornemann Lee Parker and Ted Paul

* * *

CORRECTION:
In Issue # 3 we said the 512k Ram upgrade available from Spectrum Projects was only \$49.95. This price was only good when the upgrade was purchased in conjunction with two additional software purchases. We regret any inconvenience this may have caused.

* * *

Tandy's Home Publisher for the CoCo 3 - \$39.95
Radio Shack Stores Nationwide

Requires CoCo 3 128k Minimum, one drive, printer

Tandy Home Publisher (THP) is perhaps second only to Multi-View & OS9 Level II as one of the best programs for the CoCo 3 produced by Tandy. It's an OS9 based program although you do not have to have the OS9 Level II operating system to operate it. Simply turn on your CoCo 3 and type "DOS."

The manual will instruct you just how to finish booting the program which occupies both sides of the supplied disks. I used "Backup Lightning" to backup the supplied disk onto two separate diskettes, as once installed only the "B" side of the diskette is needed. As in any situation it is always better to operate from the copies than the original.

The program will run under 128k of memory, however 512k is recommended. The extra memory will permit, if nothing else, less calls to the drive to pick up files. The program can also use a joystick, Color Mouse or the OS9 keyboard mouse. Use of the Hi-Res Joystick interface is optional. I used it with and without the interface and had no trouble in either situation. The real advantage of using the Hi-Res interface is to edit graphics images to your liking.

Let me state right here that THP is not a clone of a real desk top publisher such as Aldus Page Maker or similar product. There are too many missing features to classify this program in that class. Considering however that it costs 20 times less than many of the more complex programs, it is loaded with features which make creating a document very easy. In fact you can create ASCII text files with T/S Word or DeskMate and read them directly into the THP system. Typing text into the program is also possible, but it has limited editing functions and is not a real wp

program. THP is also compatible with Multi-View and thus can be part of a multi-tasking environment on your CoCo 3. Ah the joys of 512k of memory.

Where the program fails the desk top publishing criteria are in three major areas:

1. Lack of real type fonts. The program is supplied with computer style fonts. These number 14 and there are 14 attributes which can be assigned to the fonts to create outlines, shadow, bolds and italic variants. Point sizes of the types are also available to a limited degree ranging from 11 to 46 points. The manual does mention that larger point sizes will result in fewer characters per line. Not all point sizes are available for all fonts. The trouble here is that what is called "Bookman" isn't really very close to real Bookman or is "Helvetica" really close to real Helvetica. This is due to copyright laws governing the use of these type faces. However these "types" aren't even close to any actual types and all of the fonts have that "graphics" look to them. The most readable are the Gothic and Computer type styles. If a series of better type fonts were made available for this program it would be much improved.

2. While borders are available for the entire document, no column rules are available. Rules are the thin lines dividing columns or separating articles when continued from one page to another. Leading - the ability to adjust the width between lines of type is not available in the strictest sense. Spacing between lines is available, but this is similar to the space setting on a typewriter. Spacing is measured in pixels on the screen and pixels and point size are not the same. Large desk top publisher programs permit these functions in order to develop better page layout and continuity. THP lacks them, but with a little planning they can be overcome or worked around.

3. Background tints. On issue 3 of Clipboard we experimented with background tints. Those are the gray areas around the banner on the cover and in the lower left hand corner of the back page. Graphics layout people use a variety of tints and backgrounds to liven up a page and break up the monotony of type. There is no provision for this in THP. These tints and backgrounds are available from almost any art supply or graphics shop and could be put in place after your work was completed. It would have been nice to be able to have them on board.

CLEARBROOK SOFTWARE GROUP NEWSLETTER

WELCOME to the first issue of the Clearbrook Software Group Newsletter published in CoCo Clipboard Magazine. This is a forum for users of Clearbrook Software Group software to exchange ideas and information.

CSG IMS

CSG IMS is a powerful relational database management system for OS9 and OSK.

USING THE FIND COMMAND

The **FIND** command has a few characteristics which may be confusing to the novice CSG IMS user.

In a **FIND** statement, you need not include the value you are looking for. When the value is not included, the current key expression and field values are used to produce the value being searched for. For example, a database with a key called *name*, keyed on the expression *CAP\$(name)* could be used in this way:

```
IMS: name='jack'  
IMS: find key name  
IMS: print record, name  
17 Jake Jones
```

When the **FIND** statement is executed, the key expression is evaluated as 'JACK' (*CAP\$('jack')*). The index is then searched for this value. An exact match is not found so the next greater one is returned.

If we try the following commands:

```
IMS: find "jack"  
IMS: print record, name  
0
```

We see another characteristic of the **FIND** command. If we specify a value to find, it must conform to the type of value produced by the key expression (in this case *CAP\$(name)*). Because the index is arranged in order of evaluated key expressions, "jack" is not found. It is seen to be higher than all values in the index. For correct results use:

```
IMS: find "JACK"  
or  
IMS: find cap$("jack")
```

CHECKING FOR A UNIQUE KEY VALUE

Often when writing accounting applications, it is important to have a unique identification for an account record. This helps avoid confusion. The standard procedure before adding a record to the database is to check for a record with the same key. If it exists, an error is reported and the user allowed to enter a new value.

If you use the **FIND** statement to check for a record with the same key, the found record will be read into memory, overwriting the record you were trying to add. Obviously you want to avoid this. The solution is to use the **UNIQUE** or **DUPLICATE** functions. These functions return a **TRUE** or **FALSE** value and do not read a record into memory.

```
NOTE fields in memory were  
just ENTERed  
IF UNIQUE (KEY NAME) THEN  
  INSERT  
ELSE  
  NOTE report error  
ENDIF
```

TECHNIQUES FOR FASTER DATA ACCESS

When you need a list of records matching a selection criteria, usually the entire database is searched for matching records. If the selection criteria is directly related to a database key, you can dramatically reduce the time to find all matching records.

In a typical database a request for all records in which the name field begins with the letter M would look like:

```
IMS: LIST ALL FOR NAME BW "M"
```

All of the records in the database will be checked for a match. If there is a key called name which is for the expression *CAP\$(name)*, we can do the above list in about 1/26th of the time by using:

```
IMS: FIND KEY NAME "M"  
IMS: LIST NEXT 1000 WHILE NAME  
  BW "M"
```

The value following **NEXT** could be larger if you have more than 1000 records beginning with M. In the above example the first matching record is found very quickly with the **FIND** statement. The **WHILE** clause ensures that the program stops looking after encountering the first record for which the name does not begin with M.

NEXT ISSUE

In the next newsletter, we begin developing an accounts receivable and general ledger program using CSG IMS.

WRITE US

If you have any questions or suggestions, please write us and we will try to respond in the next newsletter.

CLEARBROOK SOFTWARE GROUP

BOX 8000 - 499

Sumas WA.

U.S.A. 98295 - 8000

BOX 8000 - 499

Abbotsford, B.C.

CANADA V2S 6H1

(604) 853-9118

WHAT IS MSF?

MSF is a file manager for OS9 level 2 which lets you use disks formatted by MS-DOS in a CoCo3. With OS9 you get three standard file managers. RBF (Random Block File) manages OS9 diskettes. SCF (Sequential Character File) manages terminals, keyboards, printers, etc.. PIPEMAN manages inter-process communication through pipes.

Because RBF does not manage the disk in the same way as MS-DOS, OS9 can not normally access an MS-DOS diskette. This is where **MSF** (MicroSoft File) comes in. It organizes a disk in the same way as MS-DOS. If you format a disk using the **MSF** manager on your CoCo3 and copy files to it, you can use those files on an MS-DOS computer. Alternatively, you can take a disk from your MS-DOS system and use it in your CoCo3. Disks are totally interchangeable between computers.

The most practical use for **MSF** is the sharing of text files between OS9 and MS-DOS computers. You can now bring word processing work home from the office to complete on your CoCo3. This newsletter was typed on an **MSF** disk using an OS9 computer, formatted using a desktop publisher on an MS-DOS computer, and output to a laser printer.

Some precautions have to be taken when using MS-DOS disks in your CoCo3. A problem occurs when you try to access an MS-DOS disk using the RBF manager. Drive /D0 and /D1 use RBF while /A and /B use **MSF**. If you have an MS-DOS disk in drive /B or /D1 (they are the same physical drive) and you issue the command DIR /D1, you can cause a system crash. Let's look at why this happens.

- RBF allocates a 256 byte buffer in memory to hold a sector of data.
- RBF requests a sector to be read.
- An MS-DOS sector is read into the buffer (512 bytes) filling the allocated memory plus the next 256 bytes as well. If that extra 256 bytes was critical to the system, it could crash!

The problem is a side effect of the disk controller chip used by the CoCo3. When you request it to read a sector, you cannot specify what size that sector should be. While the sector is being transferred to memory there is not enough time to check if too many bytes are being transferred.

You alleviate this problem through caution. Be careful to access MS-DOS disks using only the /A and /B devices. Also, make sure that the current execution and data directories are set to the correct device. If your current data directory is /D1 and you place an MS-DOS disk in drive /B, make sure you either change your data directory to a directory on /B or to the disk in physical drive 0. If you fail to do so and you accidentally access the default directory, you could crash the system.

Every month I get several inquiries wanting to know if **MSF** will allow you to run MS-DOS programs on the CoCo3. The answer to this is NO! Let's look at some of the reasons an MS-DOS emulator is not feasible for the 6809 microprocessor.

1. **Memory Management.** The 6809 can directly access only 64k of memory. Special hardware in the CoCo3 allows you to access 512k in 8k blocks. It is very expensive (in terms of processing time) to switch between memory blocks to

simulate a memory space larger than 64k. The 8088 processor used in PCs can address 640k of RAM in 64k blocks. Several 64k blocks can be addressed simultaneously with very little processor overhead.

2. **Instruction Sets.** Because the 6809 and 8088 instruction sets are completely different, you can not execute 8088 programs even if you emulate MS-DOS. You must also emulate the 8088 instruction set. For each 8088 instruction you would need a routine written in 6809 which performed the equivalent function.

CONCLUSION:

I estimate that an MS-DOS and 8088 instruction emulator would be 20 to 50 times slower than a PC.

ERINA

ERINA is a powerful 6809 machine code debugger. It is designed to make the debugging of OS9 user mode programs as painless as possible by letting you watch the execution of a program one instruction at a time.

When debugging a program which uses a static display screen or a graphics display for its output (such as a CSG IMS screen form or a game) the output of the debugger can make the program output unintelligible. This problem is easily remedied by using I/O redirection.

Suppose you have three active windows on a CoCo3:

- /term - 40 column screen
- /w1 - 80 column screen
- /w2 - 80 column screen

Activate the OS9 shell in /w1, switch to it and start **ERINA**:

OS9 : ERINA /W2 > /W2

Switch over to /w2 and, at the **ERINA** prompt, execute the program you are debugging (we will call it game).

ENTER CMD : E GAME /W1

Now continue with your debugging session. The program being debugged will appear in /w1 and **ERINA** in /w2, so by switching between the windows you can follow the progress of your program. It is recommended that any F\$!CPT service requests be disabled in the program being debugged. Then **ERINA** can receive the CTRL-E and CTRL-C abort signals.

WHY IT WORKS

ERINA does not use the standard output path (/w1). Instead it does all of its output to the standard error path (/w2). The program executed by **ERINA** inherits its standard output path (/w1) and has its input set to /w1 as well. Because **ERINA** outputs to a separate window, you can better see the progress of the program being debugged.

This same procedure will of course work on other OS9 systems which have more than one terminal. If the terminals are placed side by side, you can watch the program output on one screen and **ERINA**'s output on the other.

OS9 is a registered trademark of Microware Systems Corp.
MS-DOS is a registered trademark of Microsoft Corp.

STOCK MARKET PORTFOLIO FOR THE COLOR COMPUTER



If You Are Serious About Stocks...
This Program Is A Must!

Stock Market portfolio for the color computer will keep track of all your current stock listings and keep a listing of stocks you have sold by the year, they were sold with all totals, profit and loss, and percentages. More than one person can use this program as long as the first three letters on both first and last name are not the same. The program is menu driven and will give you the option for either screen print or information to be printed on printer.

Rush Check for \$22.00 plus \$3.00 shipping & handling to:
Paparis Enterprises
700 York St.
Williamsburg, VA 23185
Please allow 2-3 weeks for delivery
Sorry no C.O.D.S.
VA residents add 4.5% sales tax.

Now for the good points:

1. THP supports NON-TANDY printer drivers! Tandy, IBM and Epson printers (almost all Epsons will work) are supported by the program. Extensive care has been taken to let you install or change almost any of the printer parameters. The manual can be a bit fuzzy on how this is done, so be sure to read it carefully. What becomes confusing to the user are the instructions and lists of printers available for THP to use. The first part of the manual says that a list of supported printers can be found in Appendix E. Appendix E however tells you nothing about what printers are supported except that the list can be found in a THP page file called "prdriver" which can be found under the "Load Page" command in the "Command" menu. Whew, that's like going to Miami from Buffalo by way of Sacramento. And you still can't get there from here! I couldn't find "prdriver" as they said, but I stumbled across the available list under the printer options menu, under the options command. It ain't hard once you know, it's that the know'n is hard to learn. The point here is that the folks in Ft. Worth have finally accepted the fact that not everybody has a Tandy Printer, and that not everyone loves a 600 baud printer rate. You can select just about any printer rate you would like, as well as turning off the line feeds after a carriage return if your printer is set up for this. Printer flexibility with THP is a real plus.

THP works around a block oriented format. You can have either a text block or graphics block within a page section, however you can't have both occupying the same space at the same time. What that means is that if you decided to use the picture of the cat from the picture file you can't type in his name into the picture itself. You can however print it underneath his picture, or next to his picture. Text can be wrapped around the left or right side of a picture providing there is enough room to do so.

Text can also be justified left, right or full. A very nice touch is that the text can also have proportional spacing. That way the letter "i" does not take up as much room as the letter "h" and looks more natural. Text handling is a bit slow but is fun to watch as the program works the text down the columns, around the pictures and fills up the page. Text editing is next to impossible except for backspacing and re-typing. Experience using the program will be the real teacher in successfully using THP to its full potential.

There are a number of graphic pictures available for THP to use and I would suspect that more will become available either from Tandy or from users on BBS's or on CompuServe. Many of the pictures are very good, many are so-so and a couple of them are lousy. The pictures can have three sizes, small, medium and large and can be placed anywhere on a page. Eight pictures per page is the limit of the program, and not many newsletters, or news-papers will use eight pictures on a page anyways. Pictures can be edited and saved and you have good control over placement.

There are also a group of documents available to you that you can edit if you wish, or just print out. This includes various awards, notepad headings and a computer award. "Aaarggh" said Charley Brown, "The computer award looks like a Tandy Model III, not a CoCo 3." Somebody slipped here. However the awards were done pretty well and are suitable for giving to your kids for achievements in school, Sunday school, hobbies or even to your friends at work as "gag" awards after a frustrating day.

There is a lot more to say about THP than we have room for in this issue. Despite its limitations as a real desk top publisher THP provides a lot of program for the dollar. The manual could be a bit better with illustrations and a more tutorial format but it will get you started. The creativity is up to you. We recommend it.

PRODUCT REVIEW: DeskMate3

By: Paul E. Bornemann

I own and operate an Air Freight Company. Our computer system consists of a Color Computer 3, dual disk drives, Daisy Wheel 230 printer and a ton of Color Computer software. Not one single working day goes by that I don't reach for my DeskMate3 diskette.

DeskMate3 a complete workstation for the Color Computer3, hosts several different applications. You will find a calendar for scheduling, text editor for writing letters

and memos, index cards for filing information, telecom for transmitting or receiving DeskMate data files, a paint application allowing you to express your artistic talents, and my favorite, LEDGER, a complete spreadsheet program.

Two features of DM3 that I find important but not necessarily unique in today's software jungle is an 80 column display and the choice of baud rate. Now you can view 10 columns and 18 rows in spreadsheet format on a single screen. In the LEDGER you can set what size columns you would like choosing from 3 to 27 characters per column, make all the columns the same size or switch them around. You can insert mathematical functions with ease. Those of us who are familiar with the original DeskMate know the frustration of only being able to run the printer at 600 or 1200 baud. DeskMate3 now provides us with a choice of 600, 1200, 2400, 4800, and 9600 baud.

Using the LEDGER applications of the DM3 software, I keep track of my company's accounts receivable & payable. I also maintain individual account profiles on each of our clients. The following is a brief example of this report.

E.R. SQUIBB & SONS -----> ACCOUNT PROFILE PG 8 OF 8

AMOUNT	DATE	INVOICE	SHIPPER'S NAME	DATE REC'D
813.40	01/05/88	104597	C. BURGESS	PD 01/14/88
309.50	01/07/88	CV#5	B. CAZALA	PD 01/14/88
	01/07/88	105725	L. AVILA	
	01/07/88	104598	C. BURGESS	
\$1122.90	< TOTAL AMOUNT DUE BORN-AIR FREIGHT >			

CoCo Federal Tax 1987

1987 CoCo Federal Tax - \$49.95
 Puritas Springs Software
 Ameritrust Building
 17140 Lorain Ave.
 Cleveland, Ohio 44111
 (216) 251-8085

I didn't know just what to expect when Puritas Springs Software said they would be sending a copy of their 1987 CoCo Federal Tax program. Would it be one of those slap together jobs with a manual of microscopically printed pages - on 4 x 6 inch paper? Hardly! This is an excellent program and manual and I was in awe when the mail carrier handed me this huge bundle just last week.

The package consists of 3 disks and a 118 page manual with a plastic spiral binding. The binding is nice because the pages will turn totally out of the way to the back without any strain. The manual is also printed on just the right hand page which I found very pleasing as I didn't have to re-scan two pages of information after each entry session.

(Special Note: Puritas Springs Software designed this program for the small business that does tax preparation on a regular basis. The tax preparer can simply follow the program when doing a client's return and have the

Continued on page 28

Your report can vary from the very simple to the very complex, it's totally up to you.

Let me note one other advantage of DM3. When faced with all the paper work in business today, wouldn't it be great to input the information just once and then let the printer do all the work. Not having to type the same thing over and over again with the old electric typewriter is a luxury. Also, the job gets done a whole lot faster. Providing you typed the information in correctly the first time there won't be any errors to worry about.

DeskMate3 is sold at Radio Shack stores for \$99.95. It runs on OS-9 and is very easy to understand. Don't forget I just touched on the LEDGER portion of DM3, there remains a multitude of other features left to discover. It's mighty handy to have around the office and home. And it's fun!

HGRX-DUMP \$13.95

The 1ST CoCo III screen dump that prints ALL PMODE/HSCREENs to Tandy & Epson printers. Menu baud rates double-size/reverse, and 16 print patterns. Print just PMODEs on a CoCo I or II! 100% ML, disk/tape.

Sigmadisk \$14.95

Disk utility for all CoCos. Three directories, copy, backup, format, directory backup & editor, sector editor, gran table, 40/80 col. on CoCo III, more! 100% ML on disk. Supports 1-4 drives. A must!

Sigmaword \$9.95

A great low-priced word-processor for all CoCos. 50*22 screen, full-screen edit, disk I/O, 15K buffer w/ large file capability, margins, justification. 80% ML, disk only.



Ask about our games!
 No shipping charges.
 Send check or money
 order PAYABLE TO
 Kraig Brockschmidt.

SIGMA SOFTWARE
 Kraig Brockschmidt
 14024 152nd Ave. SE
 Renton, WA 98056

results for them quickly and efficiently. It also permits the preparer to store the clients data on disk in case of last minute updates. Naturally the program is perfectly suited to individual use as well. This is an extremely good example of how the CoCo can be used in business. There are certainly other tax programs for other computers, but few can match this low cost program for value. And one interesting note is that this program is used in the tax preparation business by its author!)

The program takes you through a sample session with "Jim & Betty Vance" from Cleveland. It lets you practice using the actual program and once you've run through the session with Jim & Betty, doing your own particular taxes will be much easier.

The program includes inputs and calculations for IRS forms 1040, schedules A/B/C/D/E, form 4562, 2441, 8615, worksheets for state and local income taxes, pensions, annuities, social security and railroad retirement, IRA's - just about anything you can think of that concerns your tax filing status. About the only thing I could find that the program does not handle is Part III of schedule B "Foreign Accounts and Foreign Trusts" (I don't have any) which can be handled by manual entry. The manual says that this "information does not affect the manner in which CoCo Federal Tax arrives at the tax liability."

The manual includes how to recover from a variety of problems and errors. There is no error trapping for the CoCo 3 in this version of the program as it was designed to run on any 64k machine such as the CoCo 1 & 2. Considering the low price this is not a real drawback. As in any specialized accounting or database program you are strongly encouraged to make backups of all your work. There are also several pages in the back of the manual which detail the calculation functions of the program and tells you when you may or may not take certain deductions or exemptions. It's very complete and very easy to understand.

The program will also do computer generated printouts for you. There are certain IRS regulations regarding these types of forms so be sure to check the program manual. The program comes ready to print at 9600 baud, but is adjustable for any speed you may want. It also includes a built in underline feature for the C. Itoh Prowriter. This feature is easily changed for your printer or can be disabled.

What makes using your CoCo and spending \$49.95 to do your taxes worth while? It's simple - accuracy. I.R.S. officials throughout the nation constantly remind us that more tax forms are returned for simple math errors than any other reason, except forgetting to sign the return. CoCo Federal Tax takes the worry out of the math computations. Naturally you must enter accurate information at all stages during the program and that requires having your documentation ready and have the instructions read over carefully. One of the better features of the program in this regard is a built in calculator function. This requires the use of the @ key when the command line is green and will permit you to enter

numbers and do simple addition, subtraction, multiplication and division and by pressing the = key the answer is automatically entered into the slot you just left. You can also do calculations without the auto entry feature by pressing the Q key.

There is much more to this program than I have room for in this review. At twice the price this would be a valuable asset to your program collection and will certainly take a lot of the work out of your tax preparation.

So what are you waiting for? There are only so many days before April 15th, so order your copy of CoCo Federal Tax today and get your return in the mail, and hopefully start looking for that refund check!

Inventory Manager

Inventory Manager \$25.00 (prepaid)
Forrest Enterprises
1521 Lancelot
Borger, Texas 79007

Requires 64k CoCo 1,2,3 (specify on ordering)
One drive (2 preferred), printer

If you are a small business owner or even a meticulous house keeper, Inventory Manager may be just what you need to keep track of what's on hand. This program will even write a purchase order based on your inventory needs.

The main menu allows you to:

1. Order items from different sources.
2. Adjust inventory quantities.
3. Revise item files.
4. Create new files.
5. Print item files.
6. Print inventory.
7. Sort Files.

Each menu option is a separate basic program which will allow you to return to the main menu to access other options easily.

After I read the 16 page manual, I had no trouble running the program. As I am not too business oriented, I set up a food inventory for my kitchen and printed out a grocery list. It worked fine although I had to assure my wife that this was only a temporary intrusion in her sanctuary.

I had only one question concerning the purchase order option. Why does the retail price get printed on the P.O.? It would seem to be nobody's business but my own what my markup is on an item.

My review disk had one bug in the INV MGR- program. In line 5 which checks for the number of drives in the system, the poke address should be changed from 150 to 708. Other than that, Mike Forrest did a fine job on this program. For the price, it would be a handy solution to inventory management in most small businesses.

Continued on page 29

Continued from page 10

On the Coco II, saving graphics to disk is easy; just SAVEM the contents of the screen memory. Things are more complicated under OS9 L2 on the Coco 3. The screen display requires 32K of RAM, but Basic09 doesn't know where that memory is located. It could be anywhere in the Coco 3's 512K of RAM. OS9 knows where it is, of course, but it's not talking, at least not to Basic09.

In order to save a screen to disk, it is first necessary to place the image in a graphics buffer using the GFX2 GET function. But we still don't know where the buffer is. So the procedure makes a system call which maps the buffer into our address space so we can find it. Then a User System Call is made which dumps the graphics buffer memory to a disk file. Since OS9 limits the size of a graphics buffer to 8K of RAM, this procedure must be repeated four times.

Loading a picture from disk reverses this process. A buffer is mapped into memory, then the READ User System Call is made to dump the contents of the file into the buffer. Finally the GFX2 PUT function is used to display it.

PowerGraph is, quite frankly, a hack job. I wrote it primarily to illustrate Basic09 programming techniques, the use of system calls and how to use the Coco 3's graphics capabilities. The program could have been written much more efficiently, and it lacks a lot of features necessary for a good drawing program.

PowerGraph has generated a lot of interest, however, and Ted's talked me into developing a real drawing program with all sorts of enhancements and goodies. So watch for Super PowerGraph in the near future. It will have a streamlined disk interface, saving the palette with a picture, have text functions, a much better menu system and whatever other goodies I can come up with.

Next time we're going to shift gears. The 9th Power will be on temporary vacation to make room for a piece on Extended BASIC for the Coco I/II/III, which can help make life easier for programmers and the people who use their software. Anyone who writes programs in BASIC on a hobby or professional basis will be interested. The 9th Power will be back in the sixth issue of Coco Clipboard with Deluxe PowerGraph.

If you have questions or comments, I'd be glad to hear from you. If you have hints, bug reports, patches or other information you wish to share with readers, please send them in and they'll be included in a future column. Send your letters to: Randy Krippner, 1014 W. Hwy. 114, Lot 29, Hilbert WI 54129. Please include a S.A.S.E. if you wish a reply. Disks cannot be returned unless sufficient postage and a mailer are included.

Continued on page 30

Continued from page 18

Another important point to discuss in advance of seeing the reports is the fact that there are glaring problems in the manner in which the company examines its data NOW. If you realize that DOWN TIME reported on their forms is the result of a calculation based on previously calculated fields, you will see the problem. Look again at the formula for DOWN TIME in the GIVEN DEFINITIONS section. It is the result of subtracting the STD RUN TIME from the ACT RUN TIME. STD RUN TIME is ITSELF the result of a formula that hinges on a pre-defined STD SPEED for each product. Ultimately, DOWN TIME as calculated has very little to do with ACTUAL ELAPSED TIME that occurs when the line is down. Sometimes, there may be very little difference, but other times it may be a very considerable difference.

This is the reason I added an extra field to the database. The report forms already contain an actual elapsed time for each instance of downtime on the line along with the reason. So placing a field (DOWNTIME) in the DB will at least allow the user to see the vast differences that can occur. If you don't quite understand this now, you will definitely see what I mean next issue when we look at the actual reports.

At that time, I will offer my observations and conclusions. We will look at the precise manner in which reports are setup and what is produced upon their execution. Until then, gather your questions, criticisms, and suggestions for future database applications and send them to me directly, or stay tuned 2nd Saturdays on the CoCo SIG for the Clipboard Conferences.

Continued from page 28

Vip Writer III for the CoCo 3 \$79.95
Tape Version w/o speller \$49.95
SD Enterprises
P.O. Box 1233
Gresham, Oregon 97030
(503) 663-2865

Require 128k CoCo 3, one drive, printer

VIP was, for a long time, a well known vendor name in the CoCo market. Then a couple of years ago the original VIP company pulled out of the Color Computer market place. VIP Writer was one of the best of the CoCo word processors available and I know many, many people who have used it and enjoyed it's power and flexibility.

Now that the CoCo 3 is fact it was a shame that VIP Writer would not run, consistently, on the CoCo 3. Like other word processor (wp) programs, some versions ran, and some would crash. This was frustrating to loyal users who wanted the extra features Tandy put into the CoCo 3. Enter SD Enterprises.

```

PROCEDURE chgpal
BASE 0
DIM pal(16):BYTE
DIM valid:BOOLEAN
DIM x,y,selection,counter:INTEGER
TYPE registers=cc,a,b,dp:BYTE; rx,ry,u:INTEGER
DIM regs:registers
DIM callcode:BYTE
callcode=$8E
regs.b=$91
regs.rx=ADDR(pal)
regs.a=0
RUN syscall(callcode,regs)
valid=TRUE
RUN gfx2("owset",1,0,0,10,19,2,0)
RUN gfx2("box",7,15,73,145)
RUN gfx2("box",5,12,75,147)
x=9
y=17

```

Continued from page 29

SD Enterprises has purchased the rights to the CoCo versions of the VIP Series. One of their first projects was to provide upgrades to existing VIP products to run on the CoCo 3. Our copy of VIP Calc went out to SD and was back very quickly. I booted up the program and ran perfectly on our 512k CoCo 3. Paul Anderson of SD called and said that he would be sending us a copy of VIP Writer III for the CoCo 3 to use and review, and that he wanted to point out some of the new features. The word "patch" did not cross his lips. This is not a patch, but a re-write with added features which makes using this wp a real pleasure.

What made using a wp program on the early CoCo's difficult was the lack of 80 columns. Sure you could do it with a software character set at a loss of available memory and speed, or you could drop some extra bucks for an 80 column driver. VIP Writer III takes full advantage of the CoCo 3's 40 and 80 columns screens, plus it provides 32 and 64 column screens as well. Gone are the white / black, green / black screens and the 21 or 24 line screens. VIP Writer III now provides ANY combination of 63 foreground / background colors available on the 3. For those of you with monochrome monitors VIP Writer III also provides a no color mode to avoid muddied and semi-colored displays. The program was

```

FOR counter=0 to 15
    RUN gfx2("color",counter)
    RUN gfx2("bar",x,y,x+63,y+8)
    y=y+8
NEXT counter
WHILE valid DO
    RUN readmouse(x,y)
    selection=INT(y/8)-2
    IF selection>=0 AND selection <=15 THEN
        IF pal(selection)=63 THEN
            pal(selection)=0
        ELSE
            pal(selection)=pal(selection)+1
        ENDIF
        RUN
        gfx2("palette",selection,pal(selection))
    ELSE
        valid=FALSE
    ENDIF
ENDWHILE
RUN gfx2("owend")
END

```

carefully designed to work as well on a composite monitor as on an RGB unit. We tried it on both and it was clear on sharp on the Sakata SC-100 and the Magnavox 8CM515 we used.

VIP Writer III also supports the parallel port described in the Nov. Dec. 87 issues of RAINBOW magazine and Paul now informs us that VIP Writer III will now support the J&M CP series of disk controllers which has a built in parallel printer port. This is very handy, especially if you'd like to use a printer without a serial port (laser?) and don't like another add on device to play with such as a serial to parallel converter. What else comes with the package for \$79.95?

For starters 99% of all the old VIP commands still run the same on the new program. That's a plus because you get a better program than the old one without having to learn a new command set. You get a built in 49k print spooler which holds a ton of material, but like any software spooler does slow down the output to the printer. It does not seem to bother any other function of the program however and you can stop the spooling process by pressing SHIFT and the F2 key. We used the spooler with some of the articles in this issue of Clipboard and our Epson LX-80 printer. The LX-80 we have came with Epsons serial interface card with it's own 2k buffer. VIP and the Epson had no trouble communicating and everything was printed perfectly. VIP

Writer III also uses a new cursor for the insert mode, which is a solid blinking block with the letter "I" inside. Nice touch, and you know just what is going on in this mode because of it.

So for \$79.95 you get the quality of the original VIP Writer plus the enhancements of 40 / 80 columns, choice of screen colors, or no colors, a print spooler, parallel printer availability, solid documentation and total CoCo 3 compatibility. Not a bad deal and they are shipping the product NOW. But there's a kicker, if fact two kickers, that make the VIP Writer especially value packed.

First, the package comes bundled with VIP Speller on the flip side of the master disk! VIP Speller has been patched to run on the CoCo 3, and not re-written for the 40/80 column screens. This is not a real drawback at all as you're doing a spell check and not formatting or tabular type work. The speller is the original program with the 50,000 word dictionary and the ability to add words or delete words. The program will also look at normal words, and "with numbers" words. In the first case any group of letters separated by a space is examined. In the second, any group of letters and / or numbers is checked. In any case special function markers generated by the program and ASCII codes above 129 are ignored. I love the speller and it works smoothly and picked out many of my typing fumbles. Naturally it does not correct errors in grammar, such as using "sew" for "so" or "two" for "too" or "to" or the their, there and your, you're problems often seen in typing. And you should be prepared to have a separate disk to take advantage of the speller and not get your original files mixed up with the corrected file. Naturally a second drive would be nice, a hard drive would be better. Here comes point two!

No, VIP Writer III does not come with a hard drive - darn. But it is compatible with the RGB Computer Systems RSDOS compatible Hard Drive system. In it's current configuration VIP Writer III will respond to drive requests of 0 through 3 and works perfectly with the RGB Hard drive system. In the future VIP will be compatible with the RGB system in either the 10, 20 or 40 meg configuration and the VIP command line will recognize disk commands for drives 0 through 255! This means real wp power and speed. You can print out a file via the print spooler while editing a file or loading or saving to the disk, and all at hard drive speeds. (Files sent to the spooler cannot exceed 49k and still control the computer for other tasks. If the file is greater than 49k it must be divided into small units first.) This is a real plus for CoCo users who are not, or do not want to use an OS9 based HD system. Please call SD for details on this version of the program. And please note that the program works, as far as we know, only with the RGB HD System.

Now for your \$79.95 you get everything I've mentioned and if this review sounds overly enthusiastic it is. We've been waiting for some high powered software for the CoCo 3 and VIP Writer fills the bill. Fast, easy to use and even hard drive compatibility, makes this a true bargain for the serious CoCo user.

RGB Computer Systems Hard Drive

Imagine my surprise when I picked up a copy of the competitions magazine (the one in Kentucky) and saw an ad for hard drives and the address was in my old home town of Kenmore, NY. Grandma and Grandpa live just a few blocks from Roger & Sue Krupski and so after dropping the kids at my folks, we traveled over to Stillwell Ave. and took a look at the RGB Computer Systems Hard Drive.

What makes this hard drive system different from any other for the CoCo is that it is 100% compatible with RSDOS. You do not need OS9 to run an HD using the RGB system. Roger has added several new commands to RSDOS that will permit you to have a hard drive on line with your CoCo, which looks to the CoCo like drive numbers 4 and up. In fact a 40 meg hard drive will give you the equivalent of 255 drives hooked into your Color Computer. We ran several different BASIC and machine language programs under the RGB system, including VIP Writer III, and all of them worked flawlessly. There are a few programs using ML code that have special disk I/O routines which may not work with the RGB HD as some copy protected programs may not run. Roger tells me these are very, very few and far between and in some cases can be easily patched to take advantage of the hard drive.

For those of you who are of a technical ilk here is what you get from RGB. A high quality metal case with on / off switch and a power light indicator. An 85 watt power supply is on board and is more than enough for the job. The case is designed for one or two half height drives, either floppies or an HD. A 20 meg Seagate ST-238-R drive is standard with 40 megs available. The HD controller is the ADAPTEC ACB-4099-A which is a SCSI to ST-506 / ST-412 hard disk controller. This is capable of controlling 2 hard drives. According to RGB the ADAPTEC controller does an Auto-Configure at boot up. It determines what type and size of drive you have and that means you don't have to mess with your software each time.

The RGB SCSI interface is a CoCo to SCSI host adapter using low powered Schottky chips and uses all gold connectors. RGB claims the SCSI is superior to SASI systems, in that it will permit multiple host and multiple target operations. Technically that means 2 or more CoCo's could access the hard drive. It also permits the use of newer hard drives with imbedded controllers.

The heart of the system of course are the enhancements added to RSDOS to run the HD. One change that has been made is to the DIR command. You can now specify a range value with DIR, i.e DIR 2 to 9. However with almost 255 drive available you no longer have any disk labels to look at and doing a DIR 0 to

255 would take a little too long. RGB has developed a new command called RENAME DRIVE which allows a disk "label" for each drive available. Each label can be 256 characters long! The syntax is RENAME DRIVE 9, "TERMINAL PROGRAMS: MIKEYTERM/BIN, GETERM/BIN . . ."

The Directory is also displayed in a column format. Under a CoCo 2 or 3 (yes this system is perfectly suited to run under a CoCo 2) using 32 or 40 columns you get a 2 column format. The 80 column format you get 5. The DIR also displays the "LABEL" name of the drive and the free bytes left on that drive.

Roger has also developed a further enhancement called "AUTOEXEC" which will load and run ANY program on power up. On a cold start up with the hard drive on line, the system will automatically go to drive 0 and look for a BASIC program called "AUTOEXEC/BAS". You can use this to boot up any program you'd like. I saw it work with Rogers own directory program which he wrote with a "his" & "hers" column listing. The HD will also work just dandy with your CoCo 3 in the FAST mode.

There are no set prices for the RGB Hard Drive System because each unit is custom produced. You can order the 20 or 40 meg single units, or two hard drives, or a floppy in the case. RGB will give you a quote after determining with you exactly what you would like. This is not an "off the rack" purchase, but one that is designed to meet your needs as best as possible. It's rare to be able to review an expensive piece of hardware as many places aren't going to part, even for a few weeks, with costly equipment, it was nice to go back home and see it in action.

(Note: RGB has announced that RGD-DOS is now available. This DOS will work with a floppy drive and then a hard drive later on. Please contact RGB for details.)

Continued from page 16

```

6 ****
7
10 PCLEAR1:CLEAR10000:DIMFL$(68),DT$(68)
:L$="" :WIDTH80:PALETTE0,0:PALETTE8,63:CL
S1:ATTR0,0:ON BRK GOTO880:GOTO100
20 GOSUB200:LOCATE52,22:ATTR0,6:PRINT"Dr
ive"DN"="FR"Free Granules";:LOCATE3,23:A
TTR0,6,B:PRINT">>>>";:ATTR0,6:PRINT" Cho
se - ";:ATTR0,0:PRINT" Load or Exit: <";:
ATTR0,0:PRINT" L or E";:LOCATE40,23:ATT
R0,0,B
30 PRINT"?";:ATTR0,0:LOCATE41,23:PRINT">>
";:LOCATE42,23:PRINT"";
40 EXEC44539:ON INSTR("LE",INKEY$)+1 GOT
040,700,800
98
99 *** change drive number ***
100 CLS:GOSUB400:LOCATE29,10:PRINT"-TURN
DISK DRIVE ON -":LOCATE3,23:ATTR0,6,B:
PRINT">>>>";:ATTR0,6:PRINT" Choose - ";:A
TTR0,0:PRINT" Exit or Drive Number: <E o
r 0-3 ";:ATTR0,0,B:PRINT"?";:ATTR0,0:PRI
NT">>, (ENTER=0)";:LOCATE51,23:PRINT"";

```

```

110 EXEC44539:DN=INSTR(CHR$(13)+"123E",I
NKEY$)+1:ON DN GOTO110,120,120,120,120,8
00
120 DN=DN-2:DRIVE DN:DN$=RIGHT$(STR$(DN)
,1):GOTO20*** DN$ used in LOADING ***
198
199 *** set file names in the string$ ar
ray ***
200 FR=FREE(DN):CLS:GOSUB400:LR=3:LC=0:X
1=0:X2=0
210 FOR X1=3 TO 11:DSKI$DN,17,X1,Y$,Z$:X
$=Y$
220 FOR X2=1 TO 128 STEP32:DF$=MID$(X$,X
,11):DT$=MID$(X$,X2+11,1):AN=ASC(MID$(D
F$,1,1)):IF AN=0 THEN NEXTX2:GOTO250 ELS
E IF AN=255 THEN300 ELSE C=C+1:FL$(C)=LE
FT$(DF$,8)+".":+RIGHT$(DF$,3):DT$(C)=DT$:
230 LR=LR+1:IF LR=21 THEN LR=4:LC=LC+20
240 LOCATELC,LR:PRINTFL$(C);:NEXTX2***:
don't merge ***
250 IF X$=Z$ THEN NEXTX1 ELSE X$=Z$:GOTO
220
298
299 *** clear unused array elems & check
for empty disk ***
300 FOR X=C+1 TO 68:FL$(X)="" :NEXTX:C=0:
IF FL$(1)="" THEN LOCATE31,8:PRINT"*** E
mpty Disk ***":SOUND25,2:RETURN ELSE RET
URN
399
400 LOCATE15,1:PRINT" <<< CoCo-3 Custom
Text/Color & Program Loader >>>":RETURN
498
499 *** selection (arrow) routine ***
500 LOCATE0,4:ATTR2,4:PRINTFL$(C);:LOCAT
E1,23:ATTR0,6,B:PRINT">>>>";:ATTR0,6:PRIN
T" LOAD ";:ATTR0,0:PRINT" - Select with
Arrows, Choose with Shift Up Arrow -";:C
=1:LC=0:LR=4:LOCATELC,LR:ATTR2,4:PRINTFL
$(C); 510 EXEC44539:ON
INSTR(CHR$(10)+CHR$(9)+CHR$(94)+CHR$(8)+CHR$(95),INKEY$)+1 GOTO
510,530,550,570,590,620
510 POKE341,255:POKE342,255:ON INSTR("*
+CHR$(10)+CHR$(9)+CHR$(94)+CHR$(8)+CHR$(95),INKEY$+1 GOTO510,510,530,550,570,590,
620
520 *** dw ar ***
530 GOSUB600:IF LR=20 OR LR=34 OR LR=51
OR LR=68 THEN LR=LR-16:C=C-16:GOTO610 EL
SE LR=LR+1:C=C+1:GOTO610
540 *** rt aw ***
550 GOSUB600:IF LC=60 THEN LC=0:C=C-51:G
OTO610 ELSE LC=LC+20:C=C+17:GOTO610
560 *** up aw ***
570 GOSUB600:IF LR=4 OR LR=21 OR LR=38 O
R LR=55 THEN LR=LR+16:C=C+16:GOTO610 ELS
ELR=LR-1:C=C-1:GOTO610
580 *** lf aw ***
590 GOSUB600:IF LC=0 THEN LC=60:C=C+51:G
OTO610 ELSE LC=LC-20:C=C-17:GOTO610
600 LOCATELC,LR:ATTR0,0:PRINTFL$(C);:RET
URN*** Hi-Lite Menu Selection ***

```

Continued on page 37

The Assembly Line

Kraig Brockschmidt

Welcome to the Assembly Line!

As usual I will present short machine language programs of interest and application. All programs will be given in BASIC and as a fully commented EDTASM+ source code file. All hexadecimal numbers are preceded by a \$ or &H, decimal numbers nothing.

I appreciate any response, and will personally answer questions if you include a SASE. Send your letters to:

Kraig Brockschmidt
14024 152nd. Ave. S.E.
Renton, WA 98056-7313

BASIC LINE MOVER

How much time do you spend re-typing lines of BASIC because you want to relocate them? Wouldn't a nice, simple MOVE command be great!? The program in the listing does just that: adds the command MOVE. Actually it only copies BASIC lines, but the Copy command is already used in Disk BASIC. The syntax for the command is MOVE line1, line2 where line1 is the number of the line to copy and line2 is the 'where' you want to insert it. ?FC errors occur when either line1 or line2 is greater than 63999, line1 does not exist or line2 already exists.

This utility is not perfect but it works well. BE CAUTIOUS MOVING LINES TO OR FROM THE END OF A PROGRAM. YOUR COCO MAY LOCK UP!

As always you can use either listing to enter Line Move. You can then use (C)SAVEM "LINEMOVE", &H7f1c, &H7fff, &H7F1F. To initiate the program, type CLEAR 2000, &H7F1C: LOADM "LINEMOVE": EXEC. Always EXEC Line Move after loading. This sets BASIC to recognize the MOVR command.

When you EXEC Line Move, only line 130-170 (listing 1A) are executed. We first load a from \$192, which contains the jump address for BASIC when it encounters errors. We save this original value in ERR+1, replace it with the address for Move, and return to BASIC. Now when BASIC sees a syntax error, it jumps to move. That part of the program checks if "MOVE" was entered. If not we branch to ERR, which jumps to the error drive that BASIC originally wanted.

To verify that we did enter MOVE line1-line2 and then actually move a line, we perform these steps:

- 1) Make sure "MOVE" was typed and give an error if not (430-490).
- 2) Evaluate line1 and line2 and check for a "," between them (210-420 & 500-560).
- 3) Find where to insert line2 (?FC ERROR if line2 exists), and set a flag if line2 < line1 meaning line1 is moved when it is copied (570-690).
- 4) Find where line1 starts, calculate its length and / or give and ?FC ERROR if line1 does not exist (700-860).
- 5) Check if we're only adding to the end of the program. If not, make a space to insert line2 (870-1050).
- 6) Copy line 1 to line 2 and store line2 as the line number (1060-1140).
- 7) Fix the addresses in line2 and each line after it that points to the start of the next line (i.e., add the length of line1 to each address; 1150-1260).
- 8) Take four bytes from stack to return to BASIC without errors (1270-1290).

As in the Nov / DEC issue of CoCo Clipboard, the structure of a BASIC program in memory is important to us. For each line, the structure is:

Bytes 1 & 2: address of the start of the next line.

Bytes 3 & 4: value of the line number.

Bytes 5-(up to 256) : data for the line. The last byte is always 0. After a zero, this structure repeats. The address following a zero is the address stored in bytes 1 & 2 of the previous line. These are the addresses that are fixed in step 7 above. Also \$19 contains the address for the start of the program in memory, and \$1B contains the end.

The EDTASM+ source file is well commented so I will only concentrate on the interpreting the MOVE command and the NUM subroutine. The rest of the program is mostly loops to move memory around and copy the line properly.

When you type anything in BASIC and press ENTER, it is stored in a 256 byte buffer at \$2DC. For example, suppose you type 'MOVE 45-91'. All the ASCII characters are stored in the buffer and BASIC immediately checks for a match with all its command words (i.e., PMODE, DSKINI, etc.). If there's not a match, it normally gives an ?SN ERROR. However, we intercept the error process (set up when you EXECed Line Move) and compare the first four bytes in the buffer to "MOVE" (line 430-490). If there is no match, we branch to ERR which jumps directly to BASICS error driver and returns a ?SN ERROR.

If there's a match, we calculate each line number (using NUM) and verify that there's a "," between them.

Once line1 and line2 are determined, we find the address for each line and follow the rest of the steps above. Listing 1A should be easy enough to follow from here.

The NUM subroutine is important to Line Move just as JSR \$AF67 is to BASIC. Actually, both routines are almost identical: they calculate a numerical value from ASCII digits. I am using NUM in the program 1) To demonstrate how it works and 2) because if a calculated line is >63999 and JSR \$AF67 is used the program locks up.

It is interesting exactly how the calculation is done. We first start by putting a zero in \$40 (scratchpad storage). We then get the first digit of the number and save it's value in \$42, even though we don't know how many digits the whole number has. We then load D from \$40, multiply by 10, (see lines 300-360), and add the value in \$42. Note the two commands used: ADDB \$42 and ADCA #0. The ADDB obviously adds a value to the D register. However, lets suppose B=254 and we add 6 to it. The result is B=(254+6 mod 256)=(230 mod 256)=4. Since the B register cannot hold any number >255, it "wraps-around" to 4 and the C (carry) bit in the CC register is set. To get a true addition we must add the C bit to the A register and nothing else. This is done by the ADCA #0.

The last bit I should point out is what the JSR \$9F is for in line 230. This routine (which is used extensively in the BASIC ROM) increments the address at \$A6 by one and loads the A register with the character at that address (also see line 510). It then jumps to a routine at \$AA1A which does the following:

- 1) Leaves codes \$3A-\$FF alone (no condition codes are set).
- 2) Ignores spaces (A=\$20). Jumps back to \$9F and gets the next character.
- 3) Sets the C bit if \$30 < a < \$39, i.e. A is ASCII for a number. This is why we can use BHS in line 240 to detect when A is not a number. BHS is equivalent to BCC.

I hope that the MOVE command helps your programming. Just make sure you don't do anything to lock up the computer. To be safe, always save a working copy of the program before you move any lines. One final note: BASIC seems to slow down after using MOVE,

just type CLEAR <ENTER> and things will speed back up.

Next Issue: M1. Keyboard Scanning and how to access the CTRL, ALT, F1 and F2 keys on the CoCo III.

```

10 ****
20 /* BASIC LINE MOVE */
30 /* Copyright (c)1987 */
40 /*By Kraig Brockschmidt*/
50 /*14024 152nd Ave. S.E.*/
60 /*Renton, WA 98056-7313*/
70 /* For the 'Clipboard' */
80 ****
90 CLS:CLEAR200,&H7F1C:FORT=1TO2
28:READ A$:X=X+VAL("&H"+A$):NEXT
:IF X<>24785 THEN PRINT"ERROR IN
DATA.":END
100 RESTORE:FORT=1TO228:READ A$:
POKET+&H7F1B,VAL("&H"+A$):NEXT:P
RINT"PROGRAM IN MEMORY.":END
110 DATA 7E,AC,ED,BE,01,92,AF,8C
,F8,30,8D,00,2E,BF,01,92,39,4D,4
F,56,45,FF,9E,8A,9F
120 DATA 40,9D,9F,24,1A,80,30,97
,42,DC,40,81,18,22,60,58,49,58,4
9,D3,40,58,49,DB,42
130 DATA 89,00,DD,40,20,E2,9E,40
,39,30,8C,D3,CE,02,DC,A6,C0,A1,8
0,26,B9,6D,84,2A,F6
140 DATA 33,5F,DF,A6,8D,C5,9F,44
,81,2C,26,A9,8D,BD,DE,19,33,42,A
C,C4,27,29,25,0D,33
150 DATA D8,FE,11,93,1B,25,F0,DE
,1B,73,7F,31,33,5E,34,40,9E,44,D
E,19,33,42,AC,C4,27
160 DATA 0F,25,08,33,D8,FE,11,93
,1B,25,F0,35,40,C6,08,39,30,5E,3
4,10,EC,84,A3,E4,34
170 DATA 04,DE,40,11,93,44,22,03
,3A,AF,61,DE,1B,1F,31,3A,9F,1B,7
D,7F,31,27,09,A6,C2
180 DATA A7,82,11,A3,63,22,F7,86
,FF,B7,7F,31,35,54,34,44,A6,80,A
7,C0,5A,26,F9,35,44
190 DATA 9E,40,AF,42,EF,C4,AE,C4
,3A,AF,C4,1F,13,30,02,9C,1B,25,F
3,6F,1E,6F,1F,6F,84
200 DATA 35,50,39

```

Continued on page 37

CoCo Clipboard Magazine

FOR TANDY'S
COLOR COMPUTER 2 & 3

ARTICLE LINE-UP

CoCo Club Corner
by Ted Paul

CoCo 'N Amateur Radio
by Mike Dooley

Genealogy — Program
by Ken Barber

Product Reviews
by Lee Parker and Ted Paul

Powergraph — Basic 09 Drawing Program
by Randy Krippner

The Assembly Line
by Kraig Brockschmidt

"The 9th Power"
by Randy Krippner

Pascal Programming
by Delmar Searles

On Line with CompuServe
by Wayne Day

Commentary
by Paul Bornemann

CoCo-izing Your Business
by Rush T. Caley

Is a Hard Disk in Your Future?
by Jim DeStafeno

Basic Help
by Bill Bernico

Thorns in Basic 09
by Bob van der Poel



NEW RATES

Orders Postmarked AFTER 8/1/88
1 Year of Clipboard ... \$15.00
2 years of Clipboard ... \$28.00
1 year Canadian \$18.00
(U.S. Funds)

Yes! I'm ready to join the adventure.
Enclosed is my check or money order for
\$12.00 for 6 issues (1 year) of CoCo
Clipboard Magazine.

2 years of Clipboard \$23.00

Canadian subscriptions are also \$12.00

Name _____

Address _____

City _____

State or Province _____

ZIP or Postal Code _____

Mail to: **CoCo Clipboard**

*** Magazine**

Sample issues
are \$3.50

3742 U.S. 20 Box 3. • Fredonia, N.Y. 14063

STOCK MARKET PORTFOLIO FOR THE COLOR COMPUTER

Save 10% on Stock
Market Portfolio with
this Clipboard Coupon

Paparis Enterprises
700 York St.
Williamsburg, VA 23185

MARTY'S RUBBER STAMP DISCOUNT COUPON!

We can now offer COLORS for many nylon cartridge ribbons. Our RAINBOW-PAK contain 6 different colored ribbons. Write for prices. Specify your printer's name & model.

Marty's Rubber Stamp Shop
47 South Second St.
SAINT CLAIR, PA 17970
(717) 429-1052

Check One

- SUN PRODUCTS COUPON
- FREE CATALOG
- \$5 OFF TOTAL ORDER
- -1 PER FAMILY/ MUST
- ACCOMPANY ORDER-
- EXPIRES 4/31/88

Save

20 Percent

on R. A. D.

TEXTFORM

Regularly \$34.95
Now ! --> 27.95

with this coupon
from CoCo Clipboard.

10% OFF

A complete set of Art Deli

from

Specialty
Projects

4810 MOORORY
MEMPHIS, TN
901-682-8737

EXPIRES

4/30/88



PLEASE ENCLOSE THIS COUPON WITH
YOUR ORDER SAVE \$10.00



ARK ROYAL GAMES
P.O. Box 14806
Jacksonville, FL 32238
(904) 786-8603

10% Off any Ark Royal product
in their ad with this coupon from
Clipboard!

invites you to take
5% off any 2 programs or
10% off ALL THREE programs or
advertised in this issue of Clipboard!

Send check or money orders payable to:

KRAIG BROCKSCHMIDT
Sigma Software
14024 152nd Ave SE
Renton, Wa 98056-7313

Save 10% on C.S.G.

with this coupon

Clearbrook Software Group



Information
Management
System

CSG IMS is a full featured relational database manager for the Color Computer and OS9.

- Interactive access to databases for quick queries.
- Includes a structured compiled language supporting program modules with full parameter passing and recursion.
- User defined screen and report formats.
- Record, Index, and file sizes are virtually unlimited.
- Text, BCD floating point (14 digits), short and long Integer, and date data types.

CSG IMS for CoCo2/3 OS9 L1/2 (single user) \$169.95

CSG IMS for OS9 L2 or 68000 (multi user) \$495.00

CSG IMS demo with manual \$30.00

OTHER CSG PRODUCTS:

ERINA Symbolic User-mode Debugger for OS9 \$69.00

SERINA System-mode Debugger for OS9 Level 2 \$139.00

MSF MS-DOS File Manager for CoCo3/OS9 L2 \$45.00

MSF with SDISK3 \$65.00

Shipping - N. America: \$5, Overseas: \$10

Clearbrook Software Group, Inc.
U.S.: P.O. Box 8000-499, Sunnys, WA 98295

CANADA: P.O. Box 8000-499, Abbotsford, B.C. V2S 6H1

Phone: (604)853-9118

OS9 is a trademark of Microware Systems Corp.
MS-DOS is a trademark of Microsoft Corp.



T&D Subscription
Software,
an advertiser
in this magazine,
wants to introduce you
to their variety of
inexpensive software
by giving you a:

FREE Disk or Tape
Filled with 10 programs

1 coupon per household

Circle one: TAPE DISK

Mail this coupon to:

T&D Free Offer

P.O. Box 1256
Holland, MI 49423

Continued from page 32

```

610 LOCATELC, LR:ATTR2, 4:PRINTFL$(C);:GOT
0510'*** Return Menu Item To UnHi-Lited
    Condition ***
620 RETURN
698'
699 *** load routine ***
700 L$="LOAD":DRIVE0:GOSUB500:IF FL$(C)=
    "" THEN RUN ELSE GOSUB800
710 E=ASC(DT$(C)):IF E=0 THEN LOADFL$(C)
    +"":+DN$,R ELSE IF E=2 THEN LOADM FL$(C)
    +"":+DN$
720 LOCATE0,22:ATTR0,0,B:PRINT"Selection
    Is NOT a BASIC or ML Program";:ATTR0,0:
    PRINT" Press any key."::EXEC44539:GOTO20
798'
799 *** width menu ***
800 CLS1:GOSUB400:LOCATE8,23:IF L$>"" TH
    EN ATTR0,0:PRINT">>>> Will LOAD -";CHR$(3
    4);FL$(C);CHR$(34)" <<< ";:GOTO810 ELSE
    LOCATE34,23:PRINT"EXIT to ? ";
810 ATTR0,6,B:PRINT">>>>";:ATTR0,6:PRINT"
Choose - ";:ATTR0,0:PRINT" <1, 2 or 3 ";
    :ATTR0,0,B:PRINT"?";:ATTR0,0:PRINT"> ";
820 LOCATE30,8:PRINT"(";:ATTR0,6:PRINT"1
    ";:ATTR0,0:PRINT") - 32 Char. Screen":LO
    CATE30,10:PRINT"(";:ATTR0,6:PRINT"2";:AT
    TR0,0:PRINT") - 40 Char. Screen":LOCATE3
    0,12:PRINT"(";:ATTR0,6:PRINT"3";:ATTR0,0
    :PRINT") - 80 Char. Screen":LOCATE72,23
830 EXEC44539:ON INSTR("123", INKEY$)+1 G
    OTO830,840,850,860
840 WIDTH32:PALETTE12,63:PALETTE13,0:CLS
    1:GOTO870
850 WIDTH40:GOTO870
860 GOTO870'*** remain at 80 char. ***
870 IF L$>"" THEN RETURN
880 CLS:DRIVE0:END'*** don't merge ***
899'
900'*****'
902' * CHATHAM HOUSE SOFTWARE *
903' * JAMES H. DESTAFENO *
904' * RD1, BOX 375 *
905' * WYOMING, DE 19934 *
906'*****'
907'
1000 SAVE"COLORset:0":PRINT"ok":EXEC4453
9
1010 SAVE"COLORset:1":STOP

```

Continued from page 34

LISTING 1A: EDTASM+ SOURCE CODE FILE

```

00010 ****
00020 * BASIC LINE MOVE *
00030 * Copyright (c)1987 *
00040 *By Kraig Brockschmidt*
00050 *14024 152nd Ave. S.E. *
00060 *Renton, WA 98056-7313*
00070 * For the 'Clipboard' *
00080 ****

```

```

00090
00100    ORG    $7F1C
00110    ERR    JMP    $ACED  *ACED IS A JUNK ADDRESS AND IS REPLACED
00120    *SETUP BASIC TO JUMP TO MOVE ON SYNTAX ERRORS*
00130    START  LDX    $192   *GET ADDRESS OF BASIC'S ERROR DRIVER
00140    STX    ERR+1,PCR  *STORE IT OVER ACED
00150    LEAX   MOVE,PCR  *GET ADDRESS OF MOVE
00160    STX    $192   *STORE IT AS NEW JUMP ADDRESS FOR ERRORS
00170    RTS    *RETURN TO BASIC
00180    MTX    FCC    /MOVE/ *ASCII FOR 'MOVE'
00190    FCB    $FF   *END FLAG
00200    *CALCULATE NUMBER IN X FROM ASCII STARTING AT THE ADDRESS IN $A6*
00210    NUM    LDX    $8A   *X=0
00220    STX    $40   *SCRATCHPAD STORAGE
00230    JSR    $9F   *GET NEXT CHARACTER (SEE TEXT)
00240    RHS    MOVE-3  *EXIT ROUTINE IF NEXT CHARACTER IS NOT A NUMBER
00250    SU2A   ##30  *CONVERT ASCII FOR A NUMBER TO THE NUMBER ITSELF
00260    STA    $42   *TEMPORARY SAVE
00270    LDD    $40   *GET CURRENT NUMBER VALUE
00280    CMPA   ##18  *TOP LINE # OF 63999 ($18*$FF=64000)
00290    BHI    FCERR-2 *GIVE AND ?FC ERROR IF HIGHER
00300    ASLB   ASLB  *EACH COMBINATION OF ASLB AND ROLA
00310    ROLA   ROLA  *MULTIPLIES D BY TWO. (D=D*2)
00320    ASLB   ASLB  *
00330    ROLA   ROLA  *(D=D*4)
00340    ADDD   $40  *ADD D TO D*4 WHICH EQUALS D*5
00350    ASLB   ASLB  *
00360    ROLA   ROLA  *D=D*2 AGAIN. RESULT IS D=((D*2*2)+D)*2=D*10
00370    ADDB   $42  *ADD NEXT DIGIT
00380    ADCA   $0   *ADD THE CARRY BIT TO THE MSB OF D
00390    STD    $40  *STORE NEW VALUE
00400    BRA    NUM+4 *CONTINUE
00410    LDX    $40  *GET FINAL VALUE IN X
00420    RTS    *RETURN
00430    MOVE   LEAX  MTX,PCR *POINT TO 'MOVE' TEXT
00440    LDU    $#2DC *START OF BASIC BUFFER
00450    MV1    LDA    ,U+  *GET A CHARACTER FROM BASIC BUFFER
00460    CMPA   ,X+  *COMPARE TO 'MOVE' TEXT
00470    BNE    ERR   *GIVE ERROR IF ANY DOESN'T MATCH 'MOVE'
00480    TST    ,X   *TEST NEXT BYTE IN COMPARISON TEXT
00490    RPL    MV1   *CONTINUE IF POSITIVE ($FF IS NEGATIVE)
00500    MV2    LEAU  -1,U *OTHERWISE DECREASE POINTER (WE USED LDA ,U+)
00510    STU    $A6  *STORE IT AS THE ADDRESS FOR THE $9F ROUTINE
00520    ISR    NUM   *GO GET THE FIRST LINE NUMBER=LINE1
00530    STX    $44  *STORE IT FOR LATER USE
00540    CMPA   ##2C  *CHECK FOR A ',' AFTER THE FIRST LINE NUMBER
00550    BNE    ERR   *GIVE ERROR IF NONE
00560    ISR    NUM   *GET SECOND LINE NUMBER=LINE2
00570    LDU    $19  *GET START ADDRESS OF BASIC PROGRAM
00580    CMP1   LEAU  2,U *POINT TO LINE NUMBER OF FIRST LINE IN BASIC
00590    CMPX   ,U   *COMPARE LINE2 TO LINE NUMBER AT U
00600    REQ    FCERR *GIVE ?FC ERROR IF LINE2 ALREADY EXISTS
00610    BLO    FND1  *IF LINE AT U IS HIGHER, EXIT LOOP
00620    LEAU  [-2,U] *GET ADDRESS OF NEXT BASIC LINE
00630    CMPU   $1B  *COMPARE TO END OF BASIC PROGRAM
00640    BLO    CMP1  *IF NOT DONE YET, CONTINUE COMPARISON
00650    LDU    $1B  *OTHERWISE, LOAD U WITH END OF PROGRAM ADDRESS
00660    COM    MTX+4 *SET FLAG=NO NEED TO MAKE AN INSERTION SPACE
00670    FND1  LEAU  -2,U *FIX U TO THE START ADDRESS OF THE LINE
00680    #U NOW CONTAINS THE ADDRESS WHERE WE WANT TO INSERT THE COPIED LINE*
00690    PSHS   U   *TEMPORARY STORAGE
00700    LDX    $44  *GET LINE1 FROM STORAGE
00710    LDU    $19  *GET START OF BASIC PROGRAM
00720    CMP2   LEAU  2,U *POINT TO THE LINE NUMBER
00730    CMPX   ,U   *COMPARE LINE1 TO NUMBER AT U
00740    REQ    FND2  *IF EQUAL, WE HAVE FOUND THE LINE TO COPY
00750    BLO    FCERR-2 *?FC ERROR IF LINE1 DOES NOT EXIST
00760    LEAU  [-2,U] *GET ADDRESS OF NEXT LINE
00770    CMPU   $1B  *ARE WE DONE CHECKING THE ENTIRE PROGRAM?
00780    BLO    CMP2  *NO-CONTINUE: YES-LINE1 DOES NOT EXIST
00790    PULS   U   *FIX STACK
00800    FCERR  LDB   #8  *8=CODE FOR ?FC ERROR

```

Continued on page 42

CoCo 'N Amateur Radio

Mike Dooley

Well, with Issue #3 out we should have our COCOs talking on Packet Radio. The next thing to discuss is Bulletin Boards and Digipeaters, but first I need to take care of a minor error.

In Issue #2 a layout of the pin connections between the TNC (Terminal Node Controller) and the RS-232 Program Pack were shown. These connections will work if you're using the terminal program inside the Program Pack (EXEC49152 on the COCO 2 or EXEC57360 on the COCO 3), but not with MIKEYTerm or VIDTEX. Here is a new layout that will work with MIKEYTerm, VIDTEX and the internal terminal program in the RS-232 Program Pack.

RS-232	Program Pak	TNC
2	-----	2
3	-----	3
7	-----	7
6	-.	
8	-	
20	-----	20

Wired as shown in the above layout, with pins 6, 8 and 20 tied together on the Program Pack end, the cable will work.

These same connections will work on a COCO 3. If you don't have an RS-232 Program Pack, use the following connections between the Serial Port of the COCO 2 (or 3) and the TNC:

Serial	Port	TNC
2	3	
3	7	
4	2	

When using the Serial Port you must use some kind of Terminal Program. I've used both MIKEYTerm 4.0 (COCO 2) and 4.7 (COCO 3) with the Serial Port and RS-232 Program Pack with no problems. The 80 columns on the COCO 3 is WONDERFUL!

Next let's talk about Bulletin Boards (BBS). Most of you know them as computers you can connect to via the telephone lines. They have many things available such as general and specific message bases (like the COCO SIG on Compuserve) and you can upload or download programs or information from them. BBSs can also be used as a form of electronic mail. One

such BBS system called FIDONet (sp?) allows you to send messages to other FIDONet nodes. For instance, if you're a member of a node in the Dallas area and have a friend in Boston who also is on a FIDONet node, you can send messages to each other (there is a small charge for this service). The FIDONet nodes simply go off line during the wee hours of the morning and call each other long distance to exchange the messages.

Bulletin Boards on Packet Radio do all of those same things. They have message bases, you can upload and download programs and information (usually discouraged because of the amount of time it takes) and you can send messages around the world to your friends (no charge). The difference is they don't use phone lines. Everything is done using radios and the TNC.

During most hours of operation the Bulletin Boards are found on Two Meters (usually 145.01 MHz). Sometime during the night they automatically disconnect from the Two Meter frequency and connect themselves to a frequency in the 20 Meter band (around 14100 KHz). Other Bulletin Boards do the same and messages are exchanged not just around the country, but around the world. The Japanese even have a satellite that has a Bulletin Board on it that can be used to send and receive messages!

What about Digipeaters? Well, let's say you know of a BBS you want to connect to, but your radio doesn't have enough power to reach it. You know, kind of like hollering at someone. The further away they are, the harder it is to talk to them. If you can't reach the BBS directly, but know of a Digipeater between you and the BBS, you can use it as a repeater! You connect to the BBS by using the digipeater to relay the messages back and forth.

You probably are way ahead of me by now, but let's look at the commands for the TNC to connect directly to a BBS and then to connect via a Digipeater. The BBS will be WA5MWD, a locally famous BBS mainly because it's been around so long. The Digipeater will be TI

Continued on page 40

CoCo Clipboard Magazine

OBTAINING BACK ISSUES

Back issues of CoCo Clipboard are available. They can be purchased for \$2.50 each. There is a \$1.00 postage and handling charge for the first issue ordered and \$.50 for each additional magazine that is purchased.

Vol. 1 Issue 1 Sept./Oct. 1987

TABLE OF CONTENTS

HDIR by Bob van der Poel
"The 9th Power" by Randy Krippner
Pascal Programming by Delmar Searles
"The Assembly Line" by Kraig Brockschmidt
On Line with CompuServe by Wayne Day
CoCo 'N Amateur Radio by Mike Dooley
Is a Hard Disk in Your Future? by Jim DeStafeno
CoCo-izing Your Business by Rush T. Caley

Vol. 1 Issue 3 Jan./Feb. 1988

TABLE OF CONTENTS

From the Desk of by Ted and Darlene Paul
Basic 09 Menu Maker by Bob van der Poel
Database Tutorial Pt. 1 by Rush Caley
Genealogy — Program by Ken Barber
Product Reviews by Lee Parker and Ted Paul
Powergraph — Basic 09 Drawing Program by Randy Krippner
BASIC Programming Contest by Bill Bernico et al.
CoCo Club Corner by Ted Paul
CoCo 'N Amateur Radio by Mike Dooley
Self Correcting Forecasting by Jim DeStafeno



**MISSSED AN ISSUE?
DON'T FEEL BLUE
WRITE TO US AND
WE'LL SEND IT TO YOU**

Vol. 1 Issue 2 Nov./Dec. 1987

TABLE OF CONTENTS

From The Desk of by Ted and Darlene Paul
The Assembly Line by Kraig Brockschmidt
"The 9th Power" by Randy Krippner
The Dream Machine by Jim DeStafeno
Pascal Programming by Delmar Searles
On Line with CompuServe by Wayne Day
Product Reviews by Clipboard's Review Crew
CoCo-izing Your Business by Rush T. Caley
CoCo 'N Amateur Radio by Mike Dooley
Basic Help by Bill Bernico
Thorns in Basic 09 by Bob van der Poel
Commentary by Paul Bornemann

Please send a check or money order (U.S. Funds) with your order because to keep costs down we do not bill or send C.O.D orders.

Orders may be sent to:
CoCo Clipboard Magazine
3742 U.S. 20 Box 3
Fredonia, N.Y. 14063

Continued from page 38

(another oldie here in north Dallas). Direct Connect:

C WA5MWD (CR)

Connect via the Digipeater:

C WA5MWD V TI

The 'C' means connect and the 'V' means via. What I'd like to do in future issues is list any Packet BBS's you like to use. You can send me information on your favorite BBS here at the COCO Clipboard or via Compuserve #73367, 632.

Let's look at some other things of interest. Beginning this month I'll give a short discussion and program listing covering some formula or piece of information that is helpful in getting your Amateur Radio License. This is not a full blown training course, just some helpful information.

Let's start with Resistors. These are handy devices that are used in electronic equipment to 'resist' the flow of current. They are measured in Ohms and come in a wide variety of sizes. To see how they work we'll need to learn a formula. It's called Ohms Law and goes like this:

$$E=I \cdot R$$

E is voltage (measured in volts) I is current (measured in amperes) R is resistance measured in ohms. There are two other ways to state this formula:

$$I=E/R \quad \text{and} \quad R=E/I$$

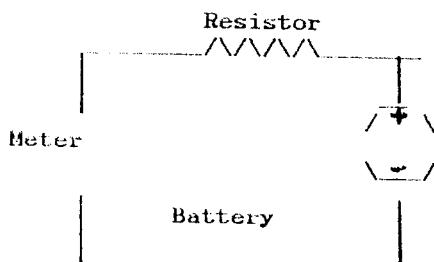
These three ways of using the same formula allow us to find any of the three values (if the other two are known) in a basic circuit. Shown in Figure 1 is our basic circuit. It contains a battery to supply voltage, a meter to measure current and a resistor to resist the flow of current. Listing 1 is a short program listing that we can use to find the unknown value in the following problems:

1.	2.	3.	4.
E=10 Volts	E=?	E=4000 Volts	E=150 Volts
I=1 Amp	I=40 Amps	I=?	I=?
R=?	R=150 Ohms	R=2000 Ohms	R=50 Ohms

What are the answers? Why not type in the listing (or, get out your calculator) and figure out the correct answers?

Next time we'll look at a piece of software for the COCO that allows us to send and receive Radioteletype (RTTY), Slow Scan TV (SSTV) and several other neat and interesting things. It's called COCORADIO and is marketed by SPEC-COM SOFTWARE, P.O. Box H, Lowden, Iowa 52255. If you send for their catalog be prepared to wait about six weeks ... they're slow, but you'll receive it. 'Till next time! 73's...

Figure 1



```

10 A$="" : E=0 : I=0 : R=0
20 PRINT "ENTER THE VALUE YOU WI
SH TO FIND: E, I OR R"
30 A$=INKEY$: IF A$="" THEN GOTO
30
40 IF A$="E" THEN GOTO 100
50 IF A$="I" THEN GOTO 150
60 IF A$="R" THEN GOTO 200
70 GOTO 10
100 PRINT:PRINT "FINDING E"
110 INPUT "I=";I
120 INPUT "R=";R
130 PRINT "E=";I*R;"VOLTS":PRINT
:PRINT
140 GOTO 10
150 PRINT:PRINT "FINDING I"
160 INPUT "E=";E
170 INPUT "R=";R
180 PRINT "I=";E/R;"AMPS":PRINT
:PRINT
190 GOTO 10
200 PRINT:PRINT "FINDING R"
210 INPUT "E=";E
220 INPUT "I=";I
230 PRINT "R=";E/I;"OHMS":PRINT
:PRINT
240 GOTO 10
    
```

BLIST for BASIC09

Bob van der Poel

I've had to put the input routine promised for this issue on hold for a little while; so, instead, we're going to discuss the various program control structures available in BASIC 09.

Program control structures you might be familiar with if you've programmed in standard BASIC include GOTO, GOSUB and FOR..NEXT loops. BASIC09 adds WHILE DO..ENDWHILE, REPEAT..UNTIL and LOOP..ENDLOOP to your programming arsenal. I don't want to rehash the BASIC09 manual which comes with OS-9 Level II, but a quick review probably won't do any of us any harm.

FOR..NEXT: This is similar to the FOR/NEXT in standard BASIC, with one important difference: The loop does not always get executed at least once. If the initial value specified as the loop counter is greater than the final value the loop will be skipped. And if you've ever wondered if using INTEGERS instead of REALs makes a difference try this program:

```
DIM counter:REAL
PRINT DATE$
FOR counter=1 to 10000
NEXT counter
PRINT DATE$
```

Now change the variable "counter" to INTEGER and see what happens . . .

GOTO: People really into structured programming frown at the use of a GOTO statement at any time, probably because it's GOTO which causes much of the spaghetti-like appearance of many BASIC programs. But there are times when you'll need to use GOTO. Don't be afraid too, but please use it sparingly.

GOSUB: Like GOTO a GOSUB must always be followed by a line number. In most cases, the initial line in the subroutine will be the only one needing a line number.

WHILE expression DO..ENDWHILE: This loop does a test to see if expression is true at the top of the loop. This means that the statements inside the loop may never be executed.

REPEAT..UNTIL expression: The only difference between this and WHILE is that the test is done at the bottom of the loop, hence the loop will always be executed at least once.

LOOP..ENDLOOP: This is essentially a "loop forever" structure. It can only be exited via an EXITIF command. This type of loop is useful when the exit conditions are set in the middle of the loop.

Rather than going into all kinds of boring and repetitive explanations of how to use these program control statements, I thought it'd be much more useful to present a non-trivial program which uses all of these statements. BLIST will take a BASIC09 which has been "printed" to a disk file and list it to the printer in a paginated form with long lines properly split. To use BLIST first type it in (don't type in the numbers in the left column), after saving a copy to disk use the PACK command to save an executable copy in your CMDs directory. Now take a program you wish to list and load it into BASIC09's workspace and type "LIST program >tempfile <ENTER>". When the disk stops spinning return to the OS-9 shell and type "BLIST <tempfile>/p <ENTER>" and watch the magic.

Apart from the various loops in the program, you should note how BLIST handles input and output. OS-9 has three standard paths it uses: Path 0 is standard input (normally your keyboard), path 1 is standard output (normally your screen) and path 2 is standard error (again, normally your screen). When you typed in "BLIST <tempfile>/p <ENTER>" you told OS-9 that you no longer wanted standard input to come from the keyboard, rather you want it to come from a file called "tempfile". In the same manner you redirected standard output to the printer (/p). Note that you did not redirect OS-9's standard error path.

BLIST sets the variables InPath, OutPath and ErrPath to 0,1 and 2. This means that input will come from the standard input -- it would be the keyboard, but you changed this when you started BLIST. This scheme lets us send BLIST's output to any device, not only the printer. We could even type in all the data the program needs -- but that would be pretty tedious . . . BLIST assumes that you have not redirected the standard error path. That's why the copyright message appears on the screen, not the printer.

So, have a good look at the program. You'll see loops within loops within loops. See if you can figure out why FOR..NEXT was used instead of WHILE DO..ENDWHILE, etc. See

Continued from page 37

```

00810 RTS
00820 FND2 LEAX -2,U *RETURN WITH ERROR
00830 PSHS X *X=START OF LINE TO COPY
00840 LDD ,X *TEMPORARY STORAGE
00850 SUBD ,S *GET ADDRESS OF NEXT LINE
00860 PSHS B *SUBTRACT FROM START ADDRESS OF THE LINE
00870 LDU $40 *B=LENGTH OF LINE TO COPY. SAVE IT ON STACK
00880 CMPU $44 *GET LINE2
00890 BHI SLTD *COMPARE TO THE LINE1
00900 ABX *IF HIGHER, SKIP NEXT TWO LINES
00910 STX 1,S *IF LOWER, WE MUST ADD LENGTH B ADDRESS OF THE
00920 *NEXT LINE SO UPON INSERTION THE ADDRESS IS
00930 *CORRECT FOR THE START OF THE NEXT LINE.#
00930 SLTD LDU $1B *GET END OF BASIC PROGRAM
00940 TFR U,X *COPY IT TO X
00950 ABX *EXTEND PROGRAM LENGTH BY B BYTES
00960 STX $1B *STORE RESULT AS NEW END
00970 TST MTX+4 *ARE WE ONLY ADDING A LINE TO THE END?
00980 BEQ MV3A *YES, SKIP NEXT 4 LINES
00990 MV3 LDA ,-U *GET A BYTE OF OLD PROGRAM
01000 STA ,-X *AND MOVE IT FORWARD B BYTES-WE OPEN A SPACE OF
01010 CMPU 3,S *B BYTES TO INSERT THE LINE WE'RE COPYING.
01020 BHI MV3 *COMPARE U WITH INSERTION POINT ADDRESS AND
01030 *CONTINUE MOVING THE PROGRAM UNTIL U IS EQUAL TO THAT ADDRESS*
01040 MV3A LDA #FF *ORIGINAL FLAG
01050 STA MTX+4 *RESTORE IT
01060 PULS B,X,U *B=LENGTH, X=START OF LINE1, U=START OF LINE2
01070 PSHS U,B *RESTORE U AND B FOR LATER USE
01080 MV4 LDA ,X+ *GET A BYTE OF THE LINE WE'RE COPYING
01090 STA ,U+ *INSERT IT IN THE NEW SPACE (OR ADD TO THE END)
01100 DECB *COUNT THE CHARACTERS
01110 BNE MV4 *CONTINUE UNTIL WE COPIED THE ENTIRE LINE
01120 PULS B,U *GET ADDRESS OF INSERTED LINE AND LENGTH
01130 LOX $40 *GET NUMBER THAT THIS LINE IS SUPPOSED TO HAVE
01140 STX 2,U *STORE IT AS THE LINE NUMBER
01150 SIU ,U *STORE U AS ADDRESS OF NEXT LINE
01160 *FIX THE 'ADDRESS OF NEXT LINE' IN INSERTED AND ALL FOLLOWING LINES*
01170 MV5 LDIX ,U *GET ADDRESS OF NEXT LINE
01180 ABX *ADD LENGTH OF INSERTED LINE
01190 STX ,U *SAVE IT BACK- NOW HAS THE CORRECT VALUE
01200 TFR X,U *PUT THAT ADDRESS IN U TO SKIP TO NEXT LINE
01210 LEAX 2,X *FIX X FOR FOLLOWING COMPARISON
01220 CMPX $1B *ARE WE DONE WITH THE PROGRAM NOW?
01230 BLO MV5 *NO-CONTINUE
01240 CLR -2,X *OTHERWISE, STORE THREE ZEROS ARE THE END OF
01250 CLR -1,X *THE PROGRAM. PREVENTS GARBAGE USING 'LIST'
01260 CLR ,X *WHICH CAN LOCK UP COMPUTER
01270 PULS X,U *FIX THE STACK TO RETURN TO BASIC WITHOUT ERROR
01280 RTS *RETURN TO BASIC
01290 END START *END OF PROGRAM

```

Continued from page 41

how loops are exited. BASIC09 helps you analyze loops with its auto-indentation.

That's it for another issue. Again, if you have ideas you'd like to see covered in future columns please write me either care of CoCo Clipboard or directly at 17435-57 Avenue, Edmonton, Alberta, Canada, T6M 1E1.

```
0000
0001      (* Blist -- BASIC09 pretty printer.
0024      (* (C) 1987, Bob van der Poel
0041
0042      (* This program will list a BASIC09 program
006D      (* or series of programs which have been
0096      (* listed to a file. Blist will paginate
```

```

(* the output and breakup long lines.
00E3
00E4
0107
0108 DIM SecondLn:BOOLEAN
010F DIM In$:STRING[500]
011B DIM Temp,Blanks:STRING[80]
012B DIM Header:STRING[40]
0137 DIM Today:STRING[20]
0143 DIM ErrPath, InPath, OutPath:BYTE
0152 DIM Indent, T, I, MaxBack:INTEGER
0165 DIM Page, LineCount, Padding:INTEGER
0174
(* Initialize a string of blanks
0195 (* to use in place of the STR$
01B3 (* function which BASIC09 lacks
01D2
01D3 Blanks=""
01DA FOR T=1 TO 80
01EA     Blanks=Blanks+" "
01F6 NEXT T
0201
0202 (* Build the current date.
021C
021D FOR T=1 TO VAL(MID$(DATE$,4,2))
0233     READ Today
0238 NEXT T
0243
0244 DATA "January", "February", "March"
0265 DATA "April", "May", "June"
027E DATA "July", "August", "September"
029E DATA "October", "November", "December"
02C2
02C3 Today=Today+" "+MID$(DATE$,7,2)+",
02E5     "+"19"+LEFT$(DATE$,2)
02E6
(* Use standard in/out.
02FD
02FE InPath=0
0305 OutPath=1
030C ErrPath=2
0313
0314 (* Let everyone know who wrote this.
0338
0339 FOR T=1 TO 6
0349     READ Temp
034E     PRINT #ErrPath USING "s78^k"; Temp
035F NEXT T
036A
036B DATA "Blist -- BASIC09 pretty printer"
0391 DATA ""
0398 DATA "(C) 1987, Bob van der Poel"
03B9 DATA "17435-57 Avenue"
03CF DATA "Edmonton, Alberta"
03E7 DATA "Canada    T6M 1E1"
03FF
0400 (* Is this a BASIC09 file?
041A
041B READ #InPath, In$
0425 IF LEFT$(In$,10)<>"PROCEDURE " THEN
043E     FOR T=1 TO 4
044E         READ Temp
0453         PRINT #ErrPath, Temp
045D NEXT T
0468
0469 END
046B
046C DATA ""
0473 DATA "This is not a BASIC09"
048F DATA "listed file, Blist aborted."
04B1 DATA ""
04B8
04B9 ENDIF
04B8
04BC (* This is the main loop,
04D5 (* continue till end of file...
04F4
04F5 WHILE NOT(EOF(#InPath)) DO
0500     Header=MID$(In$,11,LEN(In$)-10)
0512     Page=0
0519     GOSUB 100 \(* print page header
0531     GOSUB 200
0535     PRINT #OutPath, In$;
0540     PRINT #ErrPath, In$;
054A     GOSUB 200
054E     GOSUB 200
0552
0553     ^In$=""
```

```

055A
055B (* We loop here until EOF or
0577 (* a new procedure. New procedures
0599 (* loop back to the main loop and
058A (* start page numbering over.
05D7
05D8
05DA
05DB EXITIF EOF(#InPath)=TRUE THEN
05E7 ENDEXIT
05E8
05EC READ #InPath, In$
05F6
05F7 EXITIF LEFT$(In$, 10)="PROCEDURE " THEN
0610 ENDEXIT
0614
0615 (* This loop strips out any linefeeds
063A (* in the file. The blanks following
065E (* the linefeed are also deleted. This
0684 (* does not affect strings since BASIC99
06AC (* does not break up things in quotes.
06D2
06D3
06D5
06D6 T=SUBSTR(CHR$(#0A), In$)
06E3 EXITIF T=>0 THEN
06EF ENDEXIT
06F3
06F4 I=T+1
06F5
0700 WHILE MID$(In$, I, 1)=" " DO
0713 I=I+1
071E ENDWHILE
0722
0723 In$=LEFT$(In$, T-1)+MID$(In$, I, 500)
073D ENDOLOOP
0741
0742 (* Now we're ready to print the line.
0767 (* First figure the indentation level
078C (* and print the I-CODE address.
07AC
07AD
07B4 Indent=6
WHILE MID$(In$, Indent, 1)=" " AND
    Indent<LEN(In$) DO
    Indent=Indent+1
ENDWHILE
PRINT LEFT$(In$, 5);

07E9 (* Break rest of the line up into
080A (* chunks that will fit on a line.
082C (* Break up the lines at punctuation
0850 (* marks if possible.
0865
0866 In$=MID$(In$, Indent, 500)
0875 SecondLn=FALSE
087B WHILE In$>"" DO
0887     T=79-Indent
0892     IF T<LEN(In$) THEN
08A0         MaxBack=20
08A7         REPEAT
08A9             Temp=MID$(In$, T, 1)
08B7             MaxBack=MaxBack-1
08C2             T=T-1
08CD UNTIL SUBSTR(Temp, "
08F1     '/\.)*""%&'=-;+!")>0 OR MaxBack=0
08FC     T=T+1
08FE
08FF Temp=LEFT$(In$, T)
090B In$=MID$(In$, T+1, 500)
091D
091E     (* indent 2nd line 1 additional space
0943
0944     IF SecondLn=TRUE THEN
094F         Temp=" "+Temp
095B     ENDIF
095D     PRINT #OutPath, TAB(Indent); Temp;
096D     SecondLn=TRUE
0973
0974     GOSUB 200
0978     ENDWHILE
097C     ENDLOOP
0980     GOSUB 300
0984     ENDWHILE
0988     END
098A
098B 100

```

```

098F     (* start a new page
09A2
09A3 Page=Page+1
09A8 Temp=" Page "+STR$(Page)+" ... "+Today
09CB Padding=79-LEN(Temp)-LEN(Header)
09DC Temp=Temp+LEFT$(Blanks, Padding)+Header
09F0 LineCount=1
09F7 GOSUB 200
09FB PRINT #OutPath, Temp;
0A00 GOSUB 200
0A0A
0A0B 200
0A0F (* do a cr
0A19
0A1A PRINT #OutPath,
0A20 LineCount=LineCount+1
0A2B IF LineCount>=62 THEN
0A37     GOSUB 300
0A3B     GOTO 100
0A3F ENDIF
0A41 RETURN
0A43
0A44 300
0A48 (* do cr's to the end of page
0A49
0A65 FOR T=LineCount TO 66
0A77     PRINT #OutPath,
0A7D     NEXT T
0A88
0A8A

```

These better stores carry CoCo
Clipboard Magazine:

On-Line Computer Access Center
Fredonia, New York

Software Plus
Citrus Heights, Calif.

Computrac
Titusville, Florida

Mike's Electronics
Ft. Lauderdale

D&D Electronics
Angola, Indiana

Paulora Computers
Indianapolis, Indiana

Cinsoft
Cincinnati, Ohio

Personal Software
Malvern, PA

Software Corner
Reading, PA

Softhaus
Charleston Heights
South Carolina

I.O. Computers
Norfolk, Virginia

Nybbles N' Bytes
Tacoma, Washington

Cudahy New & Hobby
Cudahy, Wisconsin

REGISTRATION FORM
FOR COCO CLIPBOARD MAGAZINE'S
COCO CLUB CORNER

c/o Ted Paul
3742 U.S. 20, Box 3
Fredonia, N.Y. 14063

1. NAME OF YOUR CLUB _____

2. MAILING ADDRESS: STREET _____
CITY _____
STATE _____ ZIP (POSTAL CODE) _____

3. VOICE TELEPHONE : () _____ - _____

4. BBS TELEPHONE : () _____ - _____

5. EXACT PARAMETERS NEEDED TO ACCESS YOUR BBS:

HOURS AND DAYS OF OPERATION: MONDAY HOURS FROM _____ TO _____
(TIME ZONE) TUESDAY FROM _____ TO _____
(eastern) WEDNESDAY FROM _____ TO _____
(central) THURSDAY FROM _____ TO _____
(mountain) FRIDAY FROM _____ TO _____
(pacific) SATURDAY FROM _____ TO _____
SUNDAY FROM _____ TO _____

6. FULL OR HALF DUPLEX ? SEVEN BIT ? EIGHT BIT ? STOP BITS? _____

Do you have a regular time the BBS is off line? Yes No : When _____

7. DOES YOUR BBS SUPPORT ANY OF THE FOLLOWING AREAS:

YES NO COMMENTS

A. MESSAGE BASE -----> _____
B. UPLOADS -----> _____
C. PASSWORD NEEDED -----> _____
D. ON LINE REGISTRATION-----> _____
E. MAIL IN REGISTRATION-----> _____
F. S.I.G. SECTIONS-----> _____
G. DOWNLOAD SECTIONS-----> _____
1. GRAPHIC PROGRAM-----> _____
2. GRPAHICS PICTURES-----> _____
3. GAMES PROGRAMS-----> _____
4. ADVENTURE PROGRAMS-----> _____
5. OS-9-----> _____
6. DISK UTILITIES-----> _____
7. TELECOMMUNICATIONS PGMS-----> _____
8. OTHER-----> _____
H. ASCII DOWNLOADS-----> _____
I. XMODEM PROTOCOL-----> _____
J. KERMIT PROTOCOL-----> _____

8. DO YOU PUBLISH A NEWSLETTER YES/NO. HOW OFTEN WEEKLY, MONTHLY,
QUARTERLY

9. IS THERE A SUBSCRIPTION FEE FOR THE NEWS LETTER? Yes / No : \$ _____

10. DO YOU HAVE CLUB DUES AND IF SO HOW MUCH? _____

IN ORDER TO REGISTER YOUR CLUB FOR THE 'CORNER' WE WOULD ASK THAT IF YOU SEND
OUT A NEWSLETTER THAT YOU SEND ONE (1) SINGLE COPY TO THE ABOVE ADDRESS. WE
WILL ALSO BE CALLING YOU TO VERIFY THIS APPLICATION. IS THERE A PARTICULAR TIME
WHICH IS BEST TO CONTACT YOU ? _____ am/pm

Corrections to Program Listings
CoCo Clipboard Magazine
Vol. 1, Number 1
September / October 1987

From: Pascal Programming by Del Searles

```
program SortsDemo(input, output);

type
  DataArray = array [1..100] of integer;

var
  NumberList: DataArray;  (* Array of data elements *)
  ListLength: integer;    (* Length of data file *)

  (*****  
*  
*   Read the data file  
*  
*****)

procedure ReadFile(var NumberList: DataArray;
                    var ListLength: integer);

const
  InfileName = 'NUMBERS/DAT';  (* Physical data file *)

var
  Infile: text;  (* Logical name of data file *)

begin
  reset(Infile, InfileName);
  ListLength := 0;
  while not eof(Infile) do begin
    ListLength := ListLength + 1;
    readln(Infile, NumberList[ListLength])
  end; (* while *)
  close(Infile)
end;

  (*****  
*  
*   Print the data in the file  
*  
*****)

procedure Print(var NumberList: DataArray;
                Length: integer);

var
  LineCount, (* Number of Data elements printed *)
    (* in current line of output *)
  I: integer;    (* Loop counter *)
  Outfile: text;  (* Logical name of output file *)
```

' Corrections to Program Listings
CoCo Clipboard Magazine
Vol. 1, Number 1
September / October 1987

From: Pascal Programming by Del Searles

```
begin
  rewrite(Outfile, ':-2');
  I := 0;
  LineCount := 0;
  while I < Length do begin
    I := I + 1;
    write(Outfile, NumberList[I]:4);
    LineCount := LineCount + 1;
    if LineCount = 20 then begin
      writeln(Outfile);
      LineCount := 0
    end (* if *)
  end; (* while *)
  writeln(Outfile);
  writeln(Outfile);
  close(Outfile)
end;

(*****
*
*   Insertion Sort Routine
*
*****)

procedure InsertionSort(List: dataArray;
                        Length: integer);

var
  Temp,           (* Temporary storage for data element *)
  I, J: integer;  (* Indicate position within list *)

begin
  for I := 1 to Length-1 do begin
    J := I;
    Temp := List[I+1];
    while (Temp <= List[J]) AND (J >= 1) do begin
      List[J+1] := List[J];
      J := J - 1
    end; (* while *)
    List[J+1] := Temp
  end; (* for *)
  Print(List, Length)
end;

(*****
*
*   Selection Sort Routine
*
*****)
```

Corrections to Program Listings
CoCo Clipboard Magazine
Vol. 1, Number 1
September / October 1987

From: Pascal Programming by Del Searles

```
procedure SelectionSort(List: DataArray;  
                        Length: integer);  
  
var  
  Max,           (* Largest element *)  
  IndexOfMax,    (* Position of largest element *)  
  I, J: integer;  (* Indicate position within list *)  
  
begin  
  for I := Length downto 2 do begin  
    Max := List[1];  
    IndexOfMax := 1;  
    for J := 2 to I do  
      if List[J] > Max then begin  
        Max := List[J];  
        IndexOfMax := J  
      end; (* if *)  
    List[IndexOfMax] := List[I];  
    List[I] := Max  
  end; (* for *)  
  Print(List, Length)  
end;  
  
(*-----  
     M A I N   D R I V E R  
-----*)  
  
begin  
  ReadFile(NumberList, ListLength);  
  Print(NumberList, ListLength);  
  InsertionSort(NumberList, ListLength);  
  SelectionSort(NumberList, ListLength)  
end.
```

SPECIAL DEAL ON 500 PROGRAMS IS BACK!

BACK BY POPULAR DEMAND! GET OUR LATEST 50 DISKS OR TAPES FULL OF OVER 500 PROGRAMS. HERE IS WHAT YOU'LL RECEIVE:

- ★ Over 250 Utility/Home Application Programs including a Word Processor, Database, Spreadsheet, Disk Utilities, Business Software, Electronics Series, Educational Programs for Kids, plus much more!
- ★ Over 200 exciting games including King Pede, Kron, Star Trek, Flight Simulator, Wizard, Horse Races, Football, plus much more.
- ★ Over 30 adventures including Rambo, Haunted House, Power Sword, Skid Row, plus 32k graphic adventures.

Individual Issues sell for \$9⁰⁰ each or \$450⁰⁰ for all 50. We slashed the price to only \$150⁰⁰!

REG. \$450



\$150⁰⁰

★★THIS MONTH ONLY★★



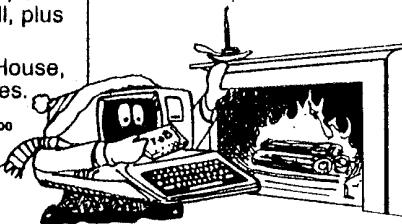
Buy this package of 500 programs and receive a free 6 month subscription.



WARM UP TO OUR WINTER PRICES ON SUBSCRIPTION SOFTWARE

THERE IS NO BETTER WAY TO ENJOY THE WINTER THAN TREATING YOUR COLOR COMPUTER TO 10 READY-TO-RUN PROGRAMS EACH MONTH. GET 12 DISKS OR TAPES A YEAR CONTAINING OVER 120 QUALITY PROGRAMS. A SUBSCRIPTION TO T & D SOFTWARE CONSISTS OF 10 READY-TO-LOAD PROGRAMS DELIVERED BY FIRST CLASS MAIL EVERY MONTH.

NO, WE ARE NOT THE SAME AS THE RAINBOW ON TAPE. IN FACT, MANY SUBSCRIBERS HAVE WRITTEN IN AND SAID THAT WE ARE MUCH BETTER THAN RAINBOW ON TAPE!



PRICES

	TAPE	THIS
	OR DISK	MONTH ONLY
1 YEAR (12 Issues)	70.00	60.00
6 MO. (6 Issues)	40.00	35.00
1 ISSUE	9.00	8.00

Michigan Residents Add 4%
Overseas Add \$10 to Subscription Price
Personal Checks Welcome!



- ★ Available on COCO 1, 2, and 3
- ★ Includes Documentation
- ★ Over 4,500 Satisfied Customers
- ★ Back Issues Available From July '82 (Over 670 Programs)

PLEASE SPECIFY TAPE OR DISK

T & D SUBSCRIPTION SOFTWARE, 2490 MILES STANDISH DR., HOLLAND, MI 49424 (616) 399-9648

CoCo Clipboard Magazine

3742 U.S. 20 Box 3, • Fredonia, N.Y. 14063

BULK MAIL
U.S. POSTAGE
PAID
PERMIT NO. 18
FREDONIA, N.Y.
14063

CoCo Clipboard Magazine

3742 U.S. 20 Box 3, Fredonia, N.Y. 14063
(716) 679-0126

Dear Clipboard Subscribers:

We have accidentally left out an entire page of Rush Caleys Data Base Tutorial from issue # 4 of the magazine.

We have made copies of these important sections for you and ask that you slip them into issue #4 so you will have a complete article.

Our apologies to you for leaving this out and our apologies to Rush as well. Rush is an excellent writer and has done an outstanding job of preparing these articles.

Ted & Darlene

DATA FIELD DESCRIPTIONS FOR DOWN TIME DATABASE

DATE:

The date of the production run for which the record is entered. This is a WORKBASE formatted field with entries to 3 elements. Day, Month, and Year. Nulls or Alpha characters will result in zeroes embedded in one of the elements.

SHIFT #:

A one character alphanumeric field that indicates what shift is represented in the record. In WORKBASE, Alphanumeric fields use only 1 byte per character. A numeric field here would use 5. (I.E. Saves room!)

PRODUCT:

The product type being run on the production line in the sample database. EG. Peanut Butter, Jelly, Soft drinks. This is a 25 character alphanumeric field.

DWNTIME TYPE:

This is a 10 character alphanumeric field that describes the type of delay involved. From our sample datafile, this might be Electrical, Mechanical, Departmental, etc.

REMARKS:

A 45 character alphanumeric field that states in brief the specific reason for the type of delay causing down time.

DEPT-CTR:

MECH-CTR:

ELEC-CTR:

These 3 fields are COUNTERS. It is an Integer field ALWAYS REQUIRING a data entry of the number 1. For each DWNTIME TYPE entered to a record, a 1 is entered to the appropriate counter field. EG a Departmental entry to the DWNTIME TYPE field would require a "1" entered to DEPT-CTR. These fields are used mainly in on-screen inquiries and tallied to a screen report which will allow the user to know the number of each type of delay from all records entered. It is a handy tool not necessary for hard-copy reports.

+:

The "+" field is not a required for the operation of the database. It is what I refer to as a "spacer" field used in hard-copy reports. WORKBASE Right justifies numeric fields and Left justifies alphanumeric fields. This can cause problems sometimes when fields on the report come to close together. So I devised the use of the "+" field of 1 byte which I can use to place spaces between reported fields.

Downtime (as distinguished from DOWN TIME)

More properly should be described as ACTUAL ELAPSED TIME during a specific delay. Each entry recording a delay requires this field as the actual number of minutes the system is down. DO NOT confuse this with the CALCULATED DOWN TIME in the section on GIVEN DEFINITIONS. (More on this in the body of the article)

STD SPEED

The STANDARD speed is a numeric field with 3 decimal places. It is the pre-determined rate of cases per minute expected to be filled during a "run". (See GIVEN DEFINITIONS)

ACT. UNITS

This field is a GIVEN for each shift. It represents the actual number of units produced by the shift in the record entered. It is an Integer field.

ACT RUN TIME

How many actual minutes elapsed to complete the number of cases filled during the run. A 3 decimal numeric field.

STD RUN TIME

A calculated field requiring no entry. STD RUN TIME = ACT. UNITS / by STD SPEED. (See GIVEN DEFINITIONS)

EFFICIENCY

A calculated field requiring no entry. EFFICIENCY = STD RUN TIME / by ACT RUN TIME * 100. (SEE GIVEN DEFINITIONS)

DOWN TIME (As distinguished from Downtime)

A calculated field requiring no entry. DOWN TIME = ACT RUN TIME - STD RUN TIME. (See GIVEN DEFINITIONS)

SHIFT START:

The starting time of the shift for the record entered. WORKBASE formatted CLOCK field. Entries to 3 elements. Hour, minutes and A (for AM) or P (for PM).

SHIFT END:

The ending time of the shift for the record is entered. See SHIFT START